HoPhage: an *ab initio* tool for identifying hosts of phage fragments from metaviromes

Jie Tan, Zhencheng Fang, Shufang Wu, Qian Guo, Xiaoqing Jiang and Huaiqiu Zhu*

State Key Laboratory for Turbulence and Complex Systems, Department of Biomedical Engineering, College of Engineering and Center for Quantitative Biology, Peking University, Beijing 100871, China.

*To whom correspondence should be addressed.

1. Introduction

Viruses are the most abundant organism on earth and phages - viruses that infect bacteria and archaea - are dominant in the virosphere (Breitbart *et al.*, 2005) and play an important role in the microbial community (Shkoporov *et al.*, 2019). To explore the ecological impact of phage in a community, it is imperative to assign the host of a given phage (de Jonge *et al.*, 2019). With the help of metagenomics technology, a wealth of novel phages that cannot be cultured are identified. Compared to the traditional culturing-based approach which naturally carries direct host information, the metagenomic method, especially metavirome, lacks the links between phages and their hosts, thus brings the increasing demand to develop computational tools for host identification of short phage fragments. However, the relevant research is still insufficient until now.

Recently, several computational strategies, mainly based on abundance profiles, genetic homology, CRISPRs, exact matches, and oligonucleotide profiles, have been proposed for host identification (Edwards *et al.*, 2016). Some of these strategies rely on a known database. For example, since some phages can integrate their genomes into host chromosomes, a prokaryote containing homologous regions with the given phage may be the potential host. Also, some prokaryotes can incorporate some DNA fragments from phages that have infected them into their own genome forming interspaced short palindromic repeats (CRISPR) spacers, hence searching the CRISPR spacers on a bacterial genome can help to identify the phage which can infect it if it contains the CRISPR system. Another strategy is based on the abundance profiles. Because phages cannot thrive without their host, the bacterium which has a good correlation in abundance with a phage across multiple samples may be the host of it. However, such approaches mentioned above present poor performance in metagenomic data. The microbial community contains a large number of novel phages that have low similarity with the known phages in the current database (Hayes *et al.*, 2017), and therefore, a similarity search-based approach cannot handle the task of host prediction of novel phages. Also, a strategy based on CRISPR spacers only works well when the phage fragment is long enough to cover the CRISPR region, hence it is not suitable for metagenomic data which contains a large number of short fragments. Moreover, a strategy based on abundance profiles requires multiple samples to calculate the correlation between each phage and bacterium, and the population dynamics between phages and their hosts often present non-linear behavior (Edwards *et al.*, 2016), increasing the difficulty of host identification.

In contrast, strategies based on sequence signatures are more suitable for metagenomic data. Phages must survive together with their host and are under strong selection pressures during the phage/host co-
Machine learning algorithms, especially deep learning algorithms, have been widely used to infer the relationship between two biological elements in the field of bioinformatics, such as gene-gene relationships (Yuan et al., 2019), protein-protein interaction (Hashemifar et al., 2018), RNA-protein interaction (Yang et al., 2018). Deep learning algorithms have been further applied to the host prediction of a virus. VIDHOPHAGE (Mock et al., 2020) is a deep learning-based virus-host prediction tool that obtains highly accurate predictions on three different virus species while using only fractions of the viral genome sequences. Markov chain model is also a popular method for researches of biological sequence. WlSH (Galiez et al., 2017) is a phage host prediction tool that uses the Markov chain model, and obtains relatively good performance on short phage fragments. The availability of the Markov chain model used by WlSH is also validated by VirHostMatcher-Net (Wang et al., 2020) since it integrates multiple features, including the score of WlSH, to predict the host of short phage fragments.

Considering the phage fragment in the metagenomic data of real community is short in length, as well as the taxonomic composition of microbial community is complex, we developed HoPhage (Host of Phage), a tool of host identification for a given phage fragment, which demonstrates high performance on short fragments within a much wider candidate host range. HoPhage integrates two modules, HoPhage-G and HoPhage-S, to improve performance, G and S respectively mean that the model is built at the genus level and the strain level. HoPhage-G is a deep learning-based module to judge whether the given phage fragment can infect any prokaryote from a specific genus. By constructing pairs of phage
fragments and prokaryotes at the genus level, host identification is transformed from a complex multi-class prediction issue to a binary classification task of judging whether there is an infection relationship between a pair. In order to improve model performance, we adopted the inception module from GoogLeNet (Szegedy et al., 2016), which can extract features at multiple scales. However, the distribution of the number of phages infecting different host genera is uneven in the current database and a large proportion of known phages derives from a narrow host range (Roux et al., 2016). As we all know, machine learning methods rely on the existing data that is used for training. So as a complement, we incorporated HoPhage-S which is a Markov chain model-based module to address this challenge. HoPhage integrates the scores of these two modules by calculating the weighted average score of them and select the host genus with the highest score as the final prediction. Testing on the benchmark dataset of artificial phage contigs and real virome data, HoPhage demonstrates superior performance on short fragments within a wide candidate host range at every taxonomic level. HoPhage is freely available at http://cqb.pku.edu.cn/ZhuLab/HoPhage/ or https://hub.docker.com/repository/docker/jietan95/hophage.

2. Material and Methods

2.1. Benchmark datasets and training-test splitting

We first downloaded 20,003 complete prokaryote genomes from the NCBI RefSeq database on May 8, 2021, including 19,629 bacterial genomes and 374 archaeal genomes. These 20,003 prokaryotic genomes range from 1,353 different genera while only 14 genomes have no genus annotation. Table “info_reseq_bacteria.csv” and “info_reseq_archaea.csv” are the information list of these prokaryote genomes and are available at http://cqb.pku.edu.cn/ZhuLab/HoPhage/data/.

We then retrieved all the 4,498 phages recorded in Virus-Host DB (updated on March 19, 2021) (Mihara et al., 2016) from NCBI. The accession number list of these phage genomes and their host annotations are shown in “info_phage_host_interaction.csv” and the distribution of host genera according to the number of phages annotated to infect them in Virus-Host DB is shown in “genus_distribution.csv”. In addition, we also added two manually verified prophage data sets that were used in our previous work PPR-Meta (Fang et al., 2019) to further supplement the phage data. After removing redundancy, 404 prophages from these two data sets were included in our data set. The relevant information of these two prophage data sets is summarized in “info_prophage_ref27_host.csv” and “info_prophage_ref42_host.csv”. These tables are also available at http://cqb.pku.edu.cn/ZhuLab/HoPhage/data/.

As an ab initio tool, it is important to evaluate whether the algorithm can identify the hosts of the fragments from a novel phage. Due to the lack of phage fragments with detailed host annotations from experimental metagenomics, we used MetaSim (Richter et al., 2008) to generate a benchmark dataset of artificial short contigs after the whole genomes of phages were downloaded from the NCBI database. For HoPhage-G, to ensure that all test data is “novel” for the model, we split the training and test set of phage genomes by the date when the phage was firstly published before we simulated artificial fragments. In general, phage genomes released before 2015 were used to train the deep learning model, and those released after 2015 were used to evaluate the algorithm. After this decisive partition, the ratios of the number of genomes in the training set to the test set of some genera were unreasonable compared with...
the traditional ratio. Therefore, we manually adjusted the genomes of phages that can infect these genera to meet a reasonable proportion of genomes in training and test data. To ensure that the training and the independent test set do not have identical or near-identical phage genomes, we followed these two principles during the adjustment process: 1) the same phage genomes will not exist in the training set and test set at the same time; 2) different phage genomes from the same study must exist in the training set or test set at the same time. Besides, the two prophage data sets were all included in the training set. For HoPhage-S, all prokaryote genomes in our dataset were used to construct the codon Markov chain model. Since there is no need to use phage genomes to construct a Markov chain model, all test set used in HoPhage-G were kept to evaluate the performance of HoPhage-S.

In the first place, three groups of phage fragments with different length intervals were constructed separately, which were 100-400 bp, 401-800 bp, and 801-1,200 bp. Since the input length of the deep learning model is fixed, we independently constructed training sets of corresponding length intervals to train the deep learning model for each group in HoPhage-G. Besides, test sets of each group were also simulated independently to evaluate the performance of HoPhage. As a result, we constructed 1,000,000 training fragments and 50,000 test fragments for each group. Subsequently, two test sets of longer phage fragments were constructed to further verify the effectiveness of HoPhage, which are 1,201-3,000 bp, 3,001-5,000 bp.

2.2. HoPhage-G: the deep learning model based module

In the existing dataset we used in this study, the number of phages that can infect a specific genus varies greatly, which can be seen from the distribution list in ‘genus_distribution.csv’. For deep learning or other machine learning algorithms, it is difficult to train a multi-classification model using a quite unbalanced dataset, especially when there is no sufficient data for some classes. It has been pointed out that the hosts of most phages have specificity at the genus level (Koskella et al., 2013). We counted all 4,843 host annotations of phages in VirusHostDB and found that only 75 (1.55%) annotations lack the genus-level information, and 403 (8.32%) annotations lack the species-level information, but 3,786 (78.17%) annotations miss strain-level information. Since it is easy to understand that the host prediction of phages can be regarded as the interaction prediction between phages and their candidate hosts, that is, to determine whether there is an infectious relationship between the phage fragment and a potential host. It is more appropriate that the pairs are constructed at the genus level for the above reasons. Therefore, in HoPhage-G, this complex multi-classification host prediction issue transformed into a two-classification task through constructing pairs of phage fragments and genera of prokaryotes. This module was named HoPhage-G because the pair was constructed at the genus level.

The process of data preparation for HoPhage-G is mainly the transformation of phage fragments and the extraction of sequence features of prokaryote genomes (Fig. S1A). For phage fragments, we first extracted artificial contigs from the phage whole genomes in the training set and the test set respectively by MetaSim. Each artificial contig was represented by “one-hot” encoding form, namely, base A, C, G, T was represented by the “one-hot” vector of [1,0,0,0], [0,1,0,0], [0,0,1,0] and [0,0,0,1]. As mentioned earlier, the length of the phage fragments in the simulated data set we constructed varies within 3 length intervals, and the input length of the deep learning model is fixed. Hence, for phage fragments whose length is shorter than Len_group (the longest length of their group), we used [0,0,0,0] to pad the end of these fragments to Len_group to make all fragments in the same group get a 4×Len_group matrix. The
Fig. S1. Data preparation for HoPhage-G and structure of deep learning neural networks in HoPhage-G. A). Constructing pairs of phage fragments and genera of prokaryotes and assigning labels to the pairs. B). Conv: Convolution neural network layer, BatchNorm: Batch normalization, FC: Fully connected layer. The numbers in Conv layer (green box) are kernel size and the number of channels. The numbers in FC layer (green box) are the input size and the output size. The six numbers in Inception module (red box) corresponding to out_ch_1, mid_ch_13, out_ch_13, mid_ch_15, out_ch_15, ch_pool_conv in dashed box on the right.

reverse complement sequences of all phage fragments were performed with the same operation to obtain other 4×Len_group matrices. These two matrices were concatenated to form the ‘Input1’, which is a 4×2Len_group matrix. Then we annotated the regions of coding sequence (CDS) for each contig from the training set through GenBank annotation. It is worth noting that for the phage fragments in the test set, considering that researchers generally do not have sufficient annotation information for the query phage when using our tool, we just used the gene prediction tool Prodigal (Hyatt et al., 2012) to annotate the CDS region of phage fragments from the test set. The CDS information of each fragment was also represented by “one-hot” encoding form, vector [1,0] and [0,1] indicates noncoding region and coding region, respectively. As before, [0,0] was used to pad this matrix and ‘Input2’ is a 2×2Len_group matrix. For prokaryotes, i.e. candidate hosts, we first clustered the prokaryote genomes by their genera, then we calculated the di-codon frequency of CDS and the 5-mer frequency of all prokaryote genomes in each cluster of different genera. The di-codon frequency of a specific genus (‘Input3’) is a 64×64 matrix and the 5-mer frequency (‘Input4’) is a 1024-dimensional vector. Finally, we constructed the pair of each
phage fragment and each candidate host genus and assigned a label to it according to the annotations of Virus-Host DB. Label 0 indicates that the pair has no infectious relationship while label 1 means that there is an infectious annotation between the phage and a prokaryote belonging to the genus in Virus-Host DB.

HoPhage-G module was built up by a deep learning model based on the inception module adopted from GoolgLeNet (Szegedy et al., 2016) and the normal convolution neural network. The structure of the deep learning model of HoPhage-G is shown in Fig. S1B. The inception module is also based on the convolution neural network, and it innovatively adopts a multi-path design, and each path uses a different convolution kernel size. Therefore, sequence features can be extracted at multiple scales by using the inception module. For each pair, HoPhage-G takes four inputs described above and outputs a score representing the possibility that the phage fragment and a prokaryote belonging to the genus have an infectious relationship.

Since the input length of the deep learning model is fixed, we independently trained the deep learning model in HoPhage-G for the three groups mentioned above. Due to the training set and the test set were separated before the phage fragments were simulated, we just respectively used the pairs constructed from genomes in the training set and the test set to train and test the model in HoPhage-G. However, a phage fragment often forms only one positive pair with one of the candidate host genera, and all other combinations are negative pairs. For the pairs used to train models in HoPhage-G, it is easy to think of an imbalance between the positive and negative samples. When there were a large number of candidate genera of hosts, this imbalance would be more serious. In our prokaryotes dataset, 1,353 genera were included. To overcome this problem, we only constructed positive and negative pairs within one mini-batch during the training process. The size of the mini-batch was set as 8, this means that the ratio of positive pairs to negative pairs during the training process has changed from 1:1,353 to 1:8. We further selected another four genera that do not exist in the minibatch to construct some additional negative pairs. These processes much help train the deep learning model. Moreover, to alleviate potential false-negative interactions, we only constructed negative pairs on the phage fragments and the genera which do not belong to the same family as the annotated host of this phage. But we still used all candidate 1,353 genera to construct pairs for each phage fragment in the test set because it is unrealistic to correctly reduce the candidate host range to eight genera in advance. During the training process, one-tenth of the training data was randomly selected as the validation set to tune the parameters in the deep learning model and determine when to stop training. After the parameters were adjusted by the validation set, all data in the training set was used to retrain the model in HoPhage-G with the same parameters.

However, it is cannot be ignored that phages with host annotation are relatively limited and many available phage genomes are distributed over a small proportion of the prokaryote genera in the current database. Since the machine learning algorithm relies much on the training dataset, it is difficult for any machine learning algorithm to handle totally novel data without any known knowledge related to it. So we could anticipate that for the results of HoPhage-G, compared with some dominant host genera, some of the genera which just have few phages annotated to infect them would have a poorer prediction performance.

2.3. HoPhage-S: the codon Markov chain model based module
To solve the above-mentioned problem that some of the genera with a small amount of data may obtain a poor prediction performance, we developed an auxiliary module HoPhage-S. Because of the genome amelioration, foreign DNA will change its sequence signatures toward its host (Suzuki et al., 2010). The Markov chain model has been widely used to measure the sequence signatures similarity between two DNA sequences. In previous related work, WIsH used the Markov chain as a mathematical model to predict the host of phages, and the ability of WIsH in predicting hosts of phages was further verified by VirHostMatcher-Net since it integrated the score of WIsH to improve performance. However, the Markov chain model in WIsH was constructed based on the base sequence. It has been pointed that most phages have to adapt to the tRNA pool of their host due to the lack of tRNA and show more consistent codon usage biases with their host (Carbone 2008). Besides, our related work showed that constructing the mathematical model based on the codon sequence was more effective than on the base sequence because the sequence signatures were more significant in the coding region (Fang et al., 2019). Considering that the CDS density of phage is much higher than that of bacteria (Amgarten et al., 2018) and the reasons mentioned above, in HoPhage-S, we trained a homogeneous codon Markov chain model for each candidate prokaryote genome using codon sequences of the CDS in it and calculated the log-likelihood of a phage fragment based on each codon Markov chain model. This module was named HoPhage-S because the Markov chain model was constructed for a single prokaryote genome (strain).

Before introducing the detailed deployment of HoPhage-S, we first explain the “base sequence” and the “codon sequence” used in it. The “base sequence” means nucleotide base sequence. For example, ATGAACGCGTAA is a base sequence. Due to the degeneracy of codons, the “codon sequence” is not equivalent to the amino acid sequence. Take the base sequence “ATGAACGCGTAA” as an example again. This sequence is actually a simplified CDS because the sequence starts with “ATG” and ends with “TAA” and the number of bases in the sequence is a multiple of 3 bp. The codon sequence of this sequence is “ATG[AAC][GCG][TAA]”. When we use $x_1$, $x_2$, $x_3$, $x_4$ represent this codon sequence, $x_1$ is “ATG”, $x_2$ is “AAC”, $x_3$ is “GCG”, and $x_4$ is “TAA”.

For each prokaryote in our candidate host dataset, the CDSs of each prokaryote genome were extracted based on the annotation information in GeneBank. We assumed $CDS_n^i$ represents the $n^{th}$ CDS region on prokaryote $i$; $N^i$ represents the total number of CDSs in prokaryote $i$; $x_1x_2...x_4$ represents a codon sequence; $#x_1x_2...x_k$ represents the number of this specific codon sequence in a certain CDS region. Then the transition probability of the codon Markov chain model for prokaryote $i$ can be represented as followed:

$$p^i(x_{k+1}|x_1...x_k) = \sum_{n=1}^{N^i} \frac{#x_1...x_{k+1}}{#x_1...x_k} \cdot CDS_n^i + 1 \cdot 64$$

where we set $k=2$ in HoPhage-S, so the transition probability of the Markov chain model is a $4096 \times 64$ matrix.

For a given phage DNA fragment to be predicted, the CDS regions were firstly extracted using Prodigal (Hyatt et al., 2012). Then the similarity score of codon sequence signatures of this query fragment which is based on the codon Markov chain model of prokaryote $i$ was calculated as:
\[
\text{score}^i = \frac{\sum_{m=1}^{M} \sum_{j=1}^{l_m-k} \log p^i(y_j^m | y_{j+k}^m \ldots y_{j+k-1}^m)}{\sum_{m=1}^{M} l_m - k}
\]

where \(y_j^m\) represents the \(j^{th}\) codon of the \(m^{th}\) CDS region on the phage fragment, \(M\) represents the total number of CDS regions on the fragment, \(l_m\) represents the number of codons on the \(m^{th}\) CDS.

### 2.4. Module integration

The whole workflow of HoPhage is shown in Fig. S2. For a query phage fragment, the CDS regions are firstly annotated by Prodigal. Then HoPhage employs HoPhage-G to score the pairs of this phage fragment and each candidate host genus in the database.

![Flowchart](image)

**Fig. S2. The flowchart of HoPhage.** HoPhage-G is first used to predict the host of given phage fragments, then how to use HoPhage-S depends on the score of HoPhage-G.

For a binary classification task, the output of the deep learning model can represent the possibility that it is a positive sample. 0.50 is the default threshold for the prediction. A higher score means that it is more likely to be a positive sample, and it also means that the prediction is more reliable. Hence, pairs
of phage fragments and candidate host genera are divided into three categories depending on the score of HoPhage-G and then different strategies are used to predict the host:

1) The highest score of all pairs in HoPhage-G (i.e. Score_Gmax) is greater than 0.80. In this case, the results of HoPhage-G are highly reliable, so in HoPhage-S we only retain the Markov chain models of prokaryotes that belong to the genera with a score higher than 0.80.

2) The Score_Gmax is between 0.40~0.80 In this case, the reliability of the results of HoPhage-G is relatively high, so in HoPhage-S we retain the Markov chain models of prokaryotes which belong to the genera with a score higher than 0.25 in HoPhage-G.

3) The Score_Gmax is less than 0.40. In this case, the reliability of the score of HoPhage-G's deep learning model is low. Therefore, in HoPhage-S we retain the Markov chain models of all prokaryotes in the existing data set.

Then the codon Markov chain models of prokaryotes retained in HoPhage-S are used to score each phage fragment, the maximum score among all Markov chain models constructed from prokaryotes that belong to the same genus will represent the score of this genus. All scores obtained by HoPhage-S are normalized to $[0, \max(\text{Score}_G, 0.5)]$. Hence, the weighted average of the highest score among one prokaryote genus in HoPhage-S and the score of this genus in HoPhage-G is set as the incorporated score for this genus. Finally, the genus with the highest score is used as the default output of HoPhage, indicating that this phage fragment is most likely to infect prokaryotes belonging to this genus.

Since the task of HoPhage is to identify hosts of short phage fragments, there are bound to deal with some short phage fragments containing only partial CDSs or no CDS on them. So here we explain how HoPhage handles these special phage fragments. For a query phage fragment, HoPhage only needs position information of the CDSs on it and does not require the CDSs to be complete. Hence, HoPhage will not perform special processing on these partial CDSs on phage fragments. The CDSs extracted by Prodigal (a gene prediction tool built in HoPhage’s software package) are arranged in the correct phase, that is, the sequence lengths of CDSs are all multiples of 3 bp. Therefore, HoPhage's processing of partial CDS is no different from that of complete CDS.

When a query phage fragment is too short, in a few cases that there is no CDS annotated by Prodigal on this phage fragment. As it is well known that phage genomes are highly optimized to be coding, this fragment is more likely to be a partial CDS that lacks 5' and 3' ends, rather than really does not contain any CDS. Since HoPhage-S needs to extract the CDSs on query phage fragments before subsequent processing when handling this kind of fragment, it will directly extract six codon sequences from the six phases of this phage DNA fragment, and each codon sequence would be served as a partial CDS to calculate the similarity score. The maximum of the six scores will be the final score of the fragment. At the same time, there is no special treatment in HoPhage-G since it will use an all-zero vector to represent this information of CDS on this phage fragment and can perform subsequent scoring.
3. Supplementary results

3.1. Evaluation of the performance of using HoPhage-G alone

We first evaluated the performance of HoPhage-G alone with a very wide host range and all 1,353 genera were used as the candidate host genera. The confusion matrices of three preliminary groups with different length intervals are shown in Fig. S3A. The upper right corner of the confusion matrix is the false positive rate (FPR), and the lower right corner is the true positive rate (TPR). It can be seen that HoPhage-G can achieve a high TPR under a very low FPR, except for the group “100-400 bp” with a relatively low TPR. To compare the performance of HoPhage-G with related tools, we regarded PHP, VirHostMatcher-Net (VHM-Net) and WlsH as binary classification tools too and draw ROC (Receiver Operating Characteristic) curve and PRC (Precision-Recall Curve) through the scoring of the model. The ROC curve and PRC with the value of AUC (Area Under Curve) and AP (Average Precision) are shown in Fig. S3B and Fig. S3C. HoPhage-G showed a clear superiority on all these evaluations. The ROC curves are commonly used to present results for binary decision problems in machine learning. The researchers have found that for the same classifier whose test set had a balanced 1:1 class distribution, when the number of negative instances has been increased 10-fold, the ROC curve almost maintained its original performance while the PRC significantly decreased (Fawcett, 2006). For HoPhage-G, the ratio of positive and negative samples is less than 1:1,000. So it can be envisioned that HoPhage-G will get a poor PRC. It also indicated that when dealing with highly skewed datasets, PRC gives a more informative picture of an algorithm's performance (Davis, et al., 2006). Whether using PRC or ROC, HoPhage-G performed better than other tools, further verified the effectiveness of the model.

It should be pointed out that since the customized candidate hosts must exist in its existing data set when using VHM-Net, we could not arbitrarily specify candidate hosts for it, so its host range was narrower than HoPhage. In detail, 1,258 genera were available in VHM-Net while HoPhage, PHP and WlsH using the consistent candidate hosts which range from 1,353 genera. This was also the same situation in the subsequent comparisons. Besides, the performance of VHM-Net is unavoidably overestimated because it integrates phage-phage similarity and we cannot remove the phages in our test set from its phage library.

Considering that in practical applications, there are not as many as 1,353 prokaryotic genera dominant in a microbial community, we limited the range of candidate hosts to 50 genera. For each comparison, we randomly selected 50 host genera from 192 genera recorded in Virus-Host DB as candidate host genera. Once the range of host genera is clarified, we would only use these genera and phage fragments to form the pairs in HoPhage-G and only keep models constructed by the prokaryotic genomes belonging to these genera in HoPhage-S, and test fragments generated from phage genomes that can infect prokaryotes belonging to one of these 50 genera were retained to evaluate model performance. Such an evaluation was repeated 20 times for each group in the test sets. For VHM-Net, in several random selections, the host range may be less than 50 due to its host range is slightly smaller than other tools. For the sake of fairness, we narrowed the host range of VHM-Net to all other available genera, and only retained phage fragments that can infect the remaining genera to calculate the prediction accuracy. The prediction accuracy was calculated as the percentage of phage fragments whose predicted hosts had the same taxonomy as their respective
annotated hosts. Fig. S4 shows the prediction accuracies of HoPhage-G and other related tools at different taxonomic levels among the host range of 50 genera. Except for HoPhage, VHM-Net was relatively the most effective tool. The prediction accuracy of PHP is the lowest on the shortest fragment, but as the length of the phage fragment increases, its accuracy gradually exceeds that of WISh. When only the genus with the highest score was treated as the prediction result, the accuracies at the genus level of HoPhage-G were 5.88%-8.92% higher than that of VHM-Net although its performance is overestimated (Fig. S4A, S4C, S4E). In other taxon levels, HoPhage-G also achieved much higher accuracies than VHM-Net. When all the genera among the top 3 scores were considered, the result of HoPhage-G was improved much more than these related tools (Fig. S4B, S4D, S4F), which indicates that HoPhage-G was more capable to give the correct result a relatively higher score than related tools.

Fig. S3. Performance of HoPhage-G and comparison with related tools. (A) Confusion matrices of HoPhage-G. ‘Inf’ means that the pair of phage fragment and host genus has an infection relationship while ‘Uninf’ means that there is no infection relationship. (B) and (C) are the receiver operating characteristic (ROC) curve and precision-recall curve (PRC) of HoPhage-G. WIsH, VirHostMatcher-Net (VHM-Net) and VirHostMatcher (VHM) on phage fragments of different lengths, respectively. The values of area under curve (AUC) and average precision (AP) are showed in legends.
Fig. S4. Prediction accuracy of HoPhage-G at different taxonomic levels within the host range of 50 genera and comparison with related tool. A, C, E are the accuracies of the top 1 prediction of host genus of HoPhage-G, WIsH, VHM-Net and VHM at each taxonomic level on 100-400, 401-800, 801-1,200 bp, respectively. B, D, F are the top 3 predictions.

3.2. Evaluation of the performance of using HoPhage-S alone

Then we also tested the performance of using module HoPhage-S alone with the same test sets and compared it with related tools. Firstly, we also regarded HoPhage-S as a two-classifier to plot the ROC curve and calculated AUC by constructing pairs at the genus level. The AUC of HoPhage-S had no advantage compared to VHM-Net and PHP (Fig. S5). As we said before, VHM-Net's performance is overestimated. Then we further compared the prediction accuracies of HoPhage-S at the different taxonomic levels with related tools among the host range of 50 genera. The result shows that the prediction accuracies of HoPhage-S were better than these two tools which do not use the information of phages (Fig. S6). Although the AUC of PHP is relatively high, the prediction accuracy at the genus level is lower than HoPhage-S. By comparing the results of the top 1 and the top 3 predictions, we found that PHP increases the accuracy more significantly when selecting multiple predictions. This shows that PHP can score the correct host relatively high, but at the same time it is difficult to score it as the highest score. Since both HoPhage-S and WIsH are based on the Markov chain model, the difference between these two is that the former uses the codon sequence of CDS to build the Markov chain model, while the latter uses the base sequence to build the model. By comparing their performance, it can be concluded that whether the signatures of the codon sequence have more potential to predict the relationship between the phage and the candidate host.

The details of the prediction accuracy of HoPhage-S and related tools at the genus levels among 20 randomly selected test sets are shown in Fig. S7. Although only using the information of the CDS region makes the sequence length that HoPhage-S could utilize becomes shorter, HoPhage-S still
**Fig. S5. ROC of HoPhage-S and comparison with related tools.** The receiver operating characteristic (ROC) curve of HoPhage-S, WisH, VirHostMatcher-Net (VHM-Net) and VirHostMatcher (VHM) on phage fragments of different lengths, respectively. The values of area under curve (AUC) are shown in legends.

**Fig. S6. Prediction accuracy of HoPhage-S at the different taxonomic levels within the host range of 50 genera and comparison with related tools.** A, C, E are the accuracies of the top 1 prediction of host genus of HoPhage-S, WisH, VHM-Net and VHM at each taxonomic level on 100–400, 401–800, 801–1200 bp, respectively. B, D, F are the results for the top 3 predictions.
Fig. S7. Prediction accuracy of HoPhage-S at the genus levels within the host range of 50 genera and comparison with related tools. A, B, C are the accuracies of the top 1 prediction of host genus of HoPhage-S, WIsH, VHM-Net and VHM at the genus level on 100-400, 401-800, 801-1200 bp, respectively. The x-axis represents different random orders.

showed better performance. This result demonstrated that, for phage, the sequence signatures extracted from the codon sequence of the coding regions are indeed more similar to the sequence signatures of their host due to greater selection pressure, hence the CDS of phage has more potential to identify hosts.

3.3. Weight selection when integrating HoPhage-G and HoPhage-S

As introduced in the manuscript, we integrated HoPhage-G and HoPhage-S by calculating the weighted average score of these two modules. As mentioned before, pairs were divided into three categories according to the HoPhage-G score (i.e. Score_Gmax) and subsequently different strategies were used to predict the host. We first counted the distribution of Score_Gmax of phage fragments, the results are shown in Table S1. As the length of the query phage fragments increased, the proportion of high scores also gradually increased, which signified that HoPhage-G can obtain more reliable prediction results for longer sequences.
To determine the best weight of these two modules, we compared the performance of HoPhage with different weights. Table S2 shows the genus accuracies when using different weight strategies to calculate the weighted average score, including uniform weights and stepped weights. In the end, the stepped weighting strategy achieved the best overall performance, which was conceivable since the level of score represents its reliability, the higher score, the better reliability. Therefore, we recommend that users choose weights according to the actual situation of their data. For example, when the sequence length is long enough, it is better to assign a larger weight to HoPhage-S. Some other suggestions are put forward in the real data application part later. In order to reduce the preference of HoPhage, we then set the weights of both modules to 0.5 to evaluate the model performance and compared it with other tools.

### Table S1. The distribution of Score\_Gmax.

| The maximum score of HoPhage-G | Score\_Gmax ≥ 0.80 | 0.80 > Score\_Gmax ≥ 0.40 | 0.40 > Score\_Gmax |
|------------------------------|-------------------|-------------------------|-------------------|
| 100-400 bp                   | 5.71%             | 69.65%                  | 24.64%            |
| 401-800 bp                   | 17.79%            | 80.49%                  | 1.73%             |
| 801-1,200 bp                 | 27.91%            | 69.11%                  | 2.98%             |

The constant weight is the weight of HoPhage-S, hence the weight of HoPhage-G is (1-weight). The ‘stepped’ means that the weight used for three categories divided by Score\_Gmax is different. S weight becomes larger as Score\_Gmax becomes smaller, which are 0.25, 0.50, and 0.75 respectively.

### Table S2. Prediction accuracy of HoPhage at the genus level with different weights.

| Weight | 0     | 0.125 | 0.25  | 0.5   | 0.75  | 0.875 | 1 stepped |
|--------|-------|-------|-------|-------|-------|-------|-----------|
| 100-400 bp | 33.33% | 34.18% | 34.93% | 36.50% | 36.35% | 34.51% | 31.54%    | 36.54%    |
| ±12.45% | ±12.45% | ±12.45% | ±12.30% | ±11.88% | ±11.34% | ±11.15% | ±12.18%   |           |
| 401-800 bp | 41.29% | 42.22% | 43.11% | 44.32% | 43.89% | 42.72% | 40.87% | 44.42%    |
| ±10.40% | ±10.40% | ±10.45% | ±10.78% | ±11.14% | ±11.18% | ±11.38% | ±10.82%   |           |
| 801-1,200 bp | 46.34% | 47.06% | 47.72% | 48.68% | 47.90% | 46.75% | 45.11% | 48.47%    |
| ±10.88% | ±10.92% | ±10.99% | ±11.31% | ±11.63% | ±11.72% | ±11.96% | ±11.42%   |           |

### 3.4. Evaluation of the performance on longer fragments

We further constructed two additional length intervals, which are 1,201-3,000 bp, 3,001-5,000 bp, 5,001-10,000 bp, and 10,001-20,000 bp, to evaluate the performance of HoPhage on longer phage fragments. As we only trained the deep learning-based model in HoPhage-G on the above three preliminary groups with the shorter input size, we first evaluated the performance of using HoPhage-G alone to test the model’s ability to handle sequences whose length is not within the preset range. For sequences longer than 1,200 bp, a scan window will move across the sequence without overlapping, and the weighted average score of all windows’ predictions is calculated. Confusion matrices (Fig. S8) of these four groups showed that HoPhage-G achieved better performance on longer phage fragments, despite using the models training by shorter fragments. The final prediction accuracies (Fig. S9) of HoPhage at every taxonomic level were also much better than all related tools.
Fig. S8. Confusion matrices of HoPhage-G on longer phage fragments. ’Inf’ means that the pair of phage fragment and host genus has an infection relationship while ‘Uninf’ means that there is no infection relationship.

Fig. S9. Prediction accuracies of HoPhage on longer phage fragments and comparison with the related tools. A, B are the accuracies of the top 1 prediction of host genus of HoPhage-S, WlsH, VHM-Net and VHM on 1200-3000, 3,000-5,000, respectively. Orange, blue, brown, green lines represent the results of HoPhage, WlsH, VHM-Net and VHM, respectively. The solid lines with error bars are the average accuracy of 20 randomly selected data. The light-colored area indicates the range of prediction accuracies.
3.5. Evaluation of the performance on phage fragments with different CDS types

Since the CDSs on a query phage fragment is the key information that HoPhage needs to use when predicting the host, and for short fragments, there may be only partial CDSs or no CDS on it, so we further explored the ability of HoPhage to predict the host of phage fragments containing different types of CDSs. The artificial phage fragments in the test set are divided into three categories based on the type of CDS on them, including “Complete”, “Partial” and “No”. The phage fragments in category “Complete” contain at least one complete CDS (may also contain partial CDSs), the phage fragments in category “Partial” only contain partial CDSs, and the phage fragments in the category “No” does not contain any CDS. Considering that some types of phage fragments are small in number, we did not separately count the accuracy of phage fragments whose host belong to randomly select 50 genera as shown in Fig. 1A in the manuscript. Hence, the prediction accuracies at the genus level in Table R1 were obtained among all the 192 host genera that have been annotated to be infected by phages in the Virus-Host DB. The results are shown in Table S3.

| Table S3. Prediction accuracy of phage fragments with different CDS types. |
|---|
|  |
| Complete | Length | Proportion | Accuracy |  |
|  |
| 100-400 |  | 1.65% | 24.84% |
| 401-800 |  | 20.23% | 31.23% |
| 801-1,200 |  | 44.23% | 35.21% |
| 1,201-3,000 |  | 75.75% | 38.94% |
| 3k-5k |  | 94.53% | 39.97% |
| 5k-10k |  | 98.81% | 42% |
| 10k-20k |  | 99.80% | 43.80% |
| Partial | Length | Proportion | Accuracy |  |
|  |
| 100-400 |  | 95.06% | 22.40% |
| 401-800 |  | 79.15% | 30.09% |
| 801-1,200 |  | 55.54% | 33.69% |
| 1,201-3,000 |  | 24.21% | 36.79% |
| 3k-5k |  | 44.23% | 40.14% |
| 5k-10k |  | 94.53% | 52.54% |
| 10k-20k |  | 98.81% | 70% |
| No | Length | Proportion | Accuracy |  |
|  |
| 100-400 |  | 3.28% | 11.66% |
| 401-800 |  | 0.63% | 11.21% |
| 801-1,200 |  | 0.22% | 13.51% |
| 1,201-3,000 |  | 0.04% | 0.00% |
| 3k-5k |  | na | na |
| 5k-10k |  | na | na |
| 10k-20k |  | na | na |

On the whole, within each group, the average lengths of the fragments in category “Complete” are the longest, followed by category “Partial”, and the shortest is category “No”. The difference in prediction accuracy between the fragments containing only partial CDSs and those containing complete CDSs is very small, which is more likely due to the length of the sequence containing only partial CDSs is slightly shorter than the fragments containing complete CDSs. However, those longer phage fragments that do not contain a CDS have a much lower host prediction accuracy than the shorter fragments that contain a complete CDS. For example, the accuracy of host prediction for phage fragments in the group “800-1,200 bp” and belonging to the category “No” (13.51%) is almost only half of that in the group “100-400 bp” and belonging to category “Complete” (24.84%). This further illustrates the importance of the CDSs in identifying the host of phage fragments and proves the rationality of our model design to pay more attention to the signatures of the CDS on phage fragments.

3.6. Evaluation of the performance on phage fragments with repetitive sequences

As far as we know, the content of repetitive sequences in the viral genome is relatively low.
Researchers have explored the relationship between virus genome size and simple sequence repeats (SSRs), and the results showed that the SSRs occurrence is strongly, significantly and positively related to the genome size (Zhao et al., 2012). They also have investigated the relative density of SSRs among differently-sized genomes. The results indicated that the genome size has slightly affected the relative density of SSRs in virus genomes and the average density of SSRs in these virus genomes is less than 3%. Therefore, we speculate that the repetitive sequences present in the phage genome will not significantly affect the predicted results of its host.

However, we still try to explore whether the repetitive sequences contained in phage fragments will affect the host prediction accuracy of HoPhage. We used the repetitive sequence annotation tool TRF (Version 4.09) (Benson 1999) to process all the artificial phage fragments in the test set. Table S4 shows the statistical information related to the repetitive sequence, including the proportion of phage fragments containing the repetitive sequence to all fragments in each group, the average length of these fragments, and the accuracy at the genus level of these fragments to predict the host. For the same considerations as above when dealing with phage fragments containing different CDS types, the prediction accuracies at the genus level in Table R2 were also obtained among all the 192 host genera that have been annotated to be infected by phages in the Virus-Host DB. As the length of the sequence increases, the proportion of fragments containing repetitive sequences also gradually increases. Among them, for phage fragments less than 1,200 bp in length, the proportion of fragments with repetitive sequences is fairly low.

Table S4. Prediction accuracy of phage fragments with repetitive sequences.

| Length (bp) | Proportion     | Repetitive Length | Accuracy  |
|------------|---------------|------------------|-----------|
|            | 100-400       | 401-800          | 801-1,200 | 1,201-3,000 | 3k-5k | 5k-10k | 10k-20k |
| 100-400    | 0.78%         | 2.22%            | 3.48%     | 7.44%       | 13.00% | 21.74% | 37.12%  |
| 401-800    | 26.36%        | 27.74%           | 31.02%    | 39.32%      | 38.91% | 42.91% | 44.37%  |

Based on the results of Table S3 and Table S4, for fragments shorter than 1,200 bp, the average lengths of the fragments containing repetitive sequences are between the fragments containing complete CDS and the fragments containing only partial CDS, but the host prediction accuracy of those fragments is slightly lower than the fragments containing only partial CDS (the host prediction accuracy of the fragments with repetitive sequences in the group “100-400 bp” is higher instead, this may be due to the fact that the proportion is too low, which leads to a preference for the results). For fragments longer than 1,200 bp, the average lengths and the host prediction accuracies of the fragments containing repetitive sequences are close to those of the fragments containing complete CDS. In summary, for short fragments, containing repetitive sequences may reduce the accuracy of predicting the host, but at the same time, the possibility that the short fragments containing repetitive sequences is also greatly reduced. For long fragments, even if the fragments contain repetitive sequences, there is still a lot of sequence information on the fragments that are useful for host identification due to the low density of repetitive sequences on the viral genome. Therefore, the repetitive sequence has little effect on HoPhage's host prediction.

3.7. Evaluation of the generalization ability of HoPhage

We further tested the generalization ability of HoPhage by eliminating phage fragments with a
high genomic similarity between the training set and the test set. For each length group, we used all
the simulated fragments in the training set as the local database, and use BLAST to eliminate the
phage fragments in the test set which have a high genomic similarity with any phage fragments in
the train set. We screened phage fragments in test sets to ensure that two phage sequences have less
than 30% of sequence identity over 60% alignment length between the training and the test set. In
the end, 40.13%, 33.78%, and 32.52% of the phage fragments in the test sets of the three groups
“100-400 bp”, “401-800 bp”, and “801-1,200 bp” meet this threshold respectively.

Table S5. Prediction accuracies of phage fragments with low genomic similarity to the training set in the test
set.

| Tool         | HoPhage | HoPhage-G | HoPhage-S | PHP     | VHM-Net | WlsH |
|--------------|---------|-----------|-----------|---------|---------|------|
|              | Top 1   | Top 1     | Top 5     | Top 1   | Top 5   | Top 1 | Top 5 |
| 100-400 bp   | 15.58%  | 12.45%    | 32.63%    | 13.89%  | 30.92%  | 6.21% | 20.65% |
| 401-800 bp   | 23.03%  | 18.75%    | 41.74%    | 20.79%  | 39.87%  | 10.85%| 30.84% |
| 801-1,200 bp | 26.60%  | 22.89%    | 48.81%    | 25.01%  | 45.22%  | 15.00%| 43.75% |

Table S5 shows the test results of the remaining phage fragments after excluding the fragments
in the test set with relatively high similarity to the fragments in the training set. Considering that the
number of remaining phage fragments is relatively small compared to the phage fragments in the
initial test set, we did not separately count the accuracy of phage fragments whose host belong to
randomly select 50 genera as shown in Fig. 1A in the manuscript. Hence, the prediction accuracies
at the genus level in Table R4 were obtained among all the 192 host genera that have been annotated
to be infected by phages in the Virus-Host DB.

Unsurprisingly, the prediction accuracy of the remaining phage fragments did decrease when
using HoPhage-G alone, however the accuracy of HoPhage-G is still higher than all other tools,
including the overestimated VHM-Net (the performance of VHM-Net is unavoidably overestimated
because it integrates phage-phage similarity and we cannot remove the phages in our test set from
its phage library when using it). And compared to WlsH and PHP, the advantage of HoPhage-G is
still remarkable.

As we mentioned before, we have considered the dependence of machine learning methods on
existing data, and that is why we designed HoPhage-S. Therefore, we are more concerned about the
overall performance of HoPhage compared with other tools. Under the cooperation of HoPhage-G
and HoPhage-S, HoPhage can even get twice the accuracy of other tools (except for VHM-Net
which has the overestimated performance on our test set). Since the prediction accuracy of all tools
has decreased, from the absolute value of the accuracy of HoPhage exceeding other tools, the
advantage of HoPhage is weakened. However, in terms of the percentage of increase in the accuracy
of HoPhage relative to other tools, HoPhage still maintains an advantage similar to the results shown
in Fig. 1A in the manuscript.

3.8. Evaluation of the performance on real data from the mock virus communities

We also used the real virome data set to evaluate the host prediction performance of HoPhage.
This data set comes from the mock virus communities which are comprised of 12 specific phages
that grow on Pseudoalteromonas, Cellulophaga baltica, and Escherichia coli (Roux et al., 2016).
The virus particles were enriched together and sequenced. We downloaded the assembled contigs from three samples, MCB1, MCB2, and MCB3, only contigs longer than 500 bp were kept in their data. Then we generated a local BLAST database with 12 phage genomes mentioned above and used blastn to trace back the source of the phage contigs, fragments with e-value less than 0.01 were kept. For the fairness of comparison, we excluded the 12 phages contained in the mock virus communities and regenerated the training data, and then retrained the deep learning model in the module HoPhage-G. The results on real virome data are introduced in the manuscript (Fig. 1B).

HoPhage experienced an expansion of the data set while improving. Before expanding the data set, the prediction accuracy of HoPhage on these three genera was significantly higher than that of WIsH. But after expanding the data set, we noticed that compared with WIsH, the advantage of HoPhage on Cellulophaga and Escherichia has been maintained, but the prediction accuracy of HoPhage in Pseudoalteromonas is almost the same as that of WIsH. In the results we got on the previous data set, HoPhage also representing more than 10% improvement on Pseudoalteromonas over WIsH. Therefore, we try to explore the reasons for this change.

When adjusting the weights of HoPhage-G and HoPhage-S, it can be observed that the prediction of the phage contigs that can infect Pseudoalteromonas in these samples is more dependent on the HoPhage-S module. Since both HoPhage-S and WIsH are Markov chain models, the difference between them is that HoPhage-S uses a Markov chain model constructed by the codon sequence of the CDSs on the genome, while WIsH uses the base sequence of the entire genome. We randomly selected some of the expanded 20,003 prokaryotes as candidate hosts and separately compared the accuracy of Pseudoalteromonas predicted by HoPhage and WIsH in different host ranges. The results are shown in Fig. S10. As the number of candidate hosts decreases, the number of strains belonging to the genus Pseudoalteromonas in the candidate hosts gradually decreases, the accuracy of HoPhage-S had almost no change, while the accuracy of WIsH dropped significantly. This phenomenon indicates that the similarity of prokaryotic strains from one genus on the CDS is higher than that of the whole genome. Therefore, when the phage-infectable prokaryotic strain P1 does not exist in the candidate hosts and the strain P2 belonging to the same genus exists in the candidate hosts, HoPhage has a greater probability of obtaining accurate host prediction than WIsH. This advantage is conducive to the research of the relationship between the prokaryotic genera that are not sufficiently studied and the phages that can infect prokaryotes belonging to them. Since the results shown in the real data section were obtained when the candidate host was all 20,003 prokaryotes expanded in this revise, the results shown in Fig. S10 also partially explain why the accuracy of HoPhage in the Pseudoalteromonas is weakened after expanding candidate hosts.

![Fig. S10. The comparison of HoPhage and WIsH among the different number of candidate hosts.](image-url)
Herein we still want to emphasize that the overall accuracy of HoPhage is only about 4% higher than WlsH. This is because the contig that can infect *Pseudoalteromonas* accounts for the majority of these real samples and the advantage of HoPhage on *Pseudoalteromonas* is weakened after we expanded the data set. But at the same time, it should be noted that compared to WlsH, HoPhage has significant advantages on these two genera *Cellulophaga* and *Escherichia* (as shown in Fig. 1B in the revised manuscript). However, because the number of phage contigs that can infect prokaryotes of these two genera is very small in these real samples, the contigs of these two genera contribute little to the overall accuracy.

In this practical application, we adjusted the scoring weights of the two modules in HoPhage according to the preliminary results. The above statistical results were finally obtained by setting the weight of HoPhage-G as 0.20 and the weight of HoPhage-S as 0.80. In this process, we found that increasing the weight of HoPhage-G can improve the prediction performance of phage contigs whose host is *Escherichia*, but at the same time, the prediction accuracy of phage contigs whose host is the other two genera decreases. When increasing the weight of HoPhage-S, the situation was just the opposite. This is probably because the volume of phages that can infect *Escherichia* is large so that the deep learning model in HoPhage-G can well summarize the relationships between these phages fragments and potential hosts and make better predictions, while the other two genera cannot obtain good predictions in deep learning model due to the relative lack of related data. Therefore, if possible, we recommend users choose appropriate weights for the two modules in HoPhage based on the community from which the phage fragments come when using HoPhage. If there are many dominant genera in this community belonging to the categories which have a large number of related records, the higher weight of module HoPhage-G may improve the prediction performance as a whole.

### 3.9. Exploration of the marker genes in phages through HoPhage-S

It has been pointed out that the evolutionary pressure of the phage genome in the co-evolution process with hosts is not uniform, and the codon usage preference will only be more prominent on some genes of phages (Carbone 2008). Since phages lack conservative genes like 16S rRNA in prokaryotes, the taxonomic classification of phages often depends on their morphology. Therefore, we assumed that based on the potential of a phage gene in identifying its host, genes that are more consistent with their host during the co-evolution can be regarded as the marker genes of phages.

As can be seen from our above results, HoPhage-S which used the codon Markov chain model to measure the similarity between phage fragments and potential hosts showed better performance than WlsH which used the base Markov chain model. In order to quantify the potential of different genes of phage in identifying the host, we used all single genes extracted from the phages in the training and test set as the inputs of HoPhage-S for host prediction. Then we calculated the accuracies of host prediction at the genus level for genes with the annotations containing different keywords. 16 keywords were used to extract the DNA sequence of annotated genes from phage genomes. According to the functions of the proteins they encode, these genes were divided into three categories, including proteins used for the processes of the central dogma (‘polymerase’, ‘ligase’, ‘primase’, ‘helicase’, ‘exonuclease’), proteins function in morphology (‘head’, ‘tail’, ‘scaffold’, ‘capsid’, ‘fiber’, ‘baseplate’) and proteins involved in the infection processes (‘lysin’,...
We used the ratio of prediction accuracy at the genus level to gene length to quantify the potential of a gene to identify the host and used the potential of genes whose annotations include "polymerase" as the reference value 1. As shown in Fig. S11, there was roughly a trend that infection-related genes had the greatest potential, morphology-related genes were the second, and the central dogma-related genes had the lowest potential. In more detail, in addition to the two genes encoding non-important structural proteins, baseplate and fiber, other morphology-related genes were slightly more potential to identify hosts than the central dogma-related genes, and the potential of infection-related genes was significantly higher than the previous two categories. This result was reasonable and in line with our expectations since the proteins involved in infection processes can directly interact with the host, which will inevitably lead to more evolutionary pressure of these relevant genes from their hosts in the co-evolution. It is worth noting that the functions of proteins ‘integrase’, ‘transposase’ and ‘excisionase’ are integration genome sequence of temperate phage into or excision it from the host chromosome (Baker 1995; Cho et al., 2002), while proteins ‘holin’ and ‘lys’ are function in cytolytic lysis process (Ugorcakova et al., 2003) which exists in both temperate and virulent phages. The results showed that genes encoding temperate phage-specific proteins had a higher potential for host identification. For example, the ‘excisionase’ gene achieved a 51.61% accuracy of host prediction at the genus level as the average length of this gene was as short as 287 bp, hence its potential in identifying the host was 18 times that of gene ‘polymerase’. This was also rational since temperate phages may integrate their genome into the host chromosome and reproduce with the host, therefore, adapting more host sequence signatures for better survival.

| keyword         | accuracy | average length of genes (bp) | number of genes | potential |
|-----------------|----------|-----------------------------|-----------------|-----------|
| polymerase      | 18.16%   | 1,697                       | 5,260           | 1         |
| ligase          | 14.70%   | 1,080                       | 1,748           | 1.27      |
| primase         | 13.70%   | 1,288                       | 2,847           | 0.99      |
| helicase        | 17.48%   | 1,430                       | 4,520           | 1.14      |
| exonuclease     | 15.83%   | 926                         | 2,211           | 1.60      |
| baseplate       | 11.73%   | 1,229                       | 4,125           | 0.89      |
| fiber           | 14.49%   | 1,802                       | 3,783           | 0.75      |
| tail            | 23.52%   | 1,239                       | 23,375          | 1.77      |
| capsid          | 29.60%   | 1,085                       | 5,406           | 2.55      |
| head            | 20.95%   | 694                         | 6,254           | 2.82      |
| scaffold        | 17.80%   | 712                         | 1,337           | 2.34      |
| lysin           | 28.25%   | 1,055                       | 3,211           | 2.50      |
| holin           | 30.30%   | 353                         | 2,307           | 8.02      |
| integrase       | 54.03%   | 1,169                       | 1,351           | 4.32      |
| transposase     | 54.09%   | 889                         | 403             | 5.69      |
| excisionase     | 51.61%   | 287                         | 93              | 16.80     |
The tetranucleotide frequency has been used to construct a phylogenetic tree of phages and found that phages with the same host converge on the tree (Pride et al., 2006). As holin exists in both temperate and virulent phages, we further constructed the phylogenetic tree of phages using the genes annotated as ‘holin’ or ‘Holin’. Fig. S12 is the phylogenetic tree constructed by these 50 holin genes with high scores when predict its hosts by HoPhage. It can be seen that genes derived from phages with the same host converge on the tree. We believed that the genes which have high potential in host identification can be used as phage marker genes for an alternative taxonomic classification of phages.
Fig. S12. Phylogenetic tree constructed by the DNA sequence of ‘holin’ genes from phages.

3.10. Comparing the running time of HoPhage and related tools

We used the sample MCB2 from the real data set to test the running time. There are 2,134
contigs in this sample. Although the total number of contigs is not very large, some of them are very long. For example, the longest contig in this sample is 961,870 bp. Therefore, the overall amount of calculation is still large enough to reflect the difference in the calculation time of each tool. However, since HoPhage was originally designed to deal with the host prediction problem of short phage fragments, we believe that the running time of short fragments is more worthy of attention. Therefore, 1,744 contigs shorter than 1,200 bp from the previously used sample were selected to test the running time of each tool again. The configuration of the computer used for the test is as follows: CPU: Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz; and Memory: 252G, DDR4; GPU: NVIDIA Tesla K40c. The running time of the four tools in these two tests is shown in Table S7. HoPhage can use GPU to accelerate the calculation of the HoPhage-G module. The batchsize we used in the test is 512, and the maximum GPU occupancy required is 2451M which is not a high GPU configuration requirement. For the fairness of comparison, we have set 4 threads for other tools if it can set multi-threading since 4 threads is also not a high configuration for a computer. The results show that the speed of PHP and WlsH is very fast, especially PHP is still very fast when dealing with very long contigs. The speed of HoPhage and VHM-Net, which need to integrate multiple information, is slower. But when dealing with short contigs, the speed of HoPhage is still much faster than that of VHM-Net while HoPhage occupies only half the memory of VHM-Net.

| Tool          | All 2,134 contigs | 1,744 short contigs |
|---------------|-------------------|---------------------|
| HoPhage       | 2h9min            | 26min               |
| PHP           | 25min             | 13min               |
| VHM-Net       | 2h18min           | 1h5min              |
| WlsH          | 1h13min           | 17min               |

Discussion

In this paper, we present HoPhage, an ab initio tool for identifying hosts of phage fragments from metavirome using the sequence signatures. Testing on a benchmark dataset of artificial short contigs and real virome data of mock viral community shows that HoPhage performs much better than the state-of-the-art tools for short fragments within a wide candidate host range. HoPhage can directly be employed on virome data, in which the viral particle is enriched before sequencing. For untargeted metagenomic data, users need to firstly identify the phage contigs from chromosome-derived contigs using related software such as VirSorter (Roux et al., 2015), VirFinder (Ren et al., 2017), PPR-Meta (Fang et al., 2019), and DeepVirFinder (Ren et al., 2020) and then use HoPhage to identify the host of the phage fragments. Like other host prediction software, users can specify the candidate host for HoPhage. In the released package, HoPhage-G contains pre-calculated di-codon frequencies and 5-mer frequencies of 1,353 genera from 20,003 prokaryotes and HoPhage-S contains all Markov chain models trained by these 20,003 prokaryotes. Under default parameters, HoPhage will use all candidate hosts in the dataset. In addition, users can also provide a list to restrict the host range for virome data from a certain environment. For example, in the human gut, about 15 bacterial genera occupy a total of 70% of organisms of the microbial community (Li et al., 2014). When employing HoPhage over gut virome data, users can restrict the candidate host within these genera to avoid false-positive prediction of a host that does not exist in this environment. On the other hand, our evaluations show that HoPhage can also achieve satisfactory performance even
if there are hundreds of candidate host genera, which means that the lack of prior knowledge about
the candidate host range will not serious affect the usage of HoPhage.

To make a reliable prediction, we designed two modules to improve HoPhage’s performance,
named HoPhage-G and HoPhage-S. HoPhage-G is a deep learning-based module. Through
constructing pairs of phage fragments and genera of potential hosts, this complex multi-
classification host prediction issue transforms into a two-classification task. Hence HoPhage-G aims
to judge whether the query phage fragment can infect a prokaryote from a specific genus. We also
adopt the inception module from GoogLeNet, which can extract features at multiple scales. While
deep learning algorithms have shown a strong ability to extract sequence signatures over a large
scale dataset to make a reliable prediction and have already been employed by many tools to predict
the interaction between biological components, HoPhage-G demonstrates superior performance as
it can obtain a high TPR under a very low FPR. However, machine learning-based tools rely on
existing data that is used for training. Since the distribution of the number of phages that can infect
hosts from different genera in the current database is unbalanced and a large number of known
phages derives from a narrow host range, therefore HoPhage-G presents a poorer performance for
some phages which lack the related data. So we designed HoPhage-S as the complement, which is
a Markov chain-based module to overcome this unbalanced problem. The innovation of HoPhage-
S is that we employed a codon Markov chain model for CDS regions in the prokaryote genomes,
rather than the base Markov chain model that WIsH employed. It has been shown that sequence
 signatures are more concentrated in the coding sequence and the density of CDS on the phage
genome is higher than that in the prokaryote genome. Our results show that the codon Markov chain
model is indeed more effective than the base Markov chain model. At last, testing on the artificial
 benchmark dataset of artificial phage contigs and real virome data, HoPhage demonstrates much
better performance on short fragments within a wide candidate host range at every taxonomic level.

However, there are still some shortcomings in our work and needed to solve in the future. It
must be admitted that although HoPhage can handle novel phages, due to the limitations of the
algorithm and currently accessible dataset, phages with very little relevant data will inevitably get
poorer prediction accuracy. In addition to sequence signatures, it is also worth considering
employing other signals to further improve the performance of HoPhage in future researches, such
as the presence of CRISPR spacers or the abundance profiles. Moreover, HoPhage is designed
primarily for the prokaryotic virus (i.e. phages), which is dominant in the microbial community, but
the real virome data may contain a small number of eukaryotic viruses. Recently, the host prediction
tool for several specific eukaryotic viruses has been designed (Mock et al., 2020). In order to let
HoPhage more versatile, it is also worth considering the host prediction for eukaryotic virus
fragments. But while most of the eukaryotic viruses are RNA viruses, which will not appear in large
amounts in DNA sequencing data, this problem has little impact on the application of HoPhage.
Another problem is that even in the virome data, there will still be some host contamination, hence
the de-hosting operation before using HoPhage will be more conducive.

In addition, the candidate hosts and query phage fragments may carry horizontally transferred
elements. Herein, we then discuss whether horizontally transferred elements will affect HoPhage’s
application and what impact it will have from two aspects. The first aspect is about the horizontally
transferred elements on the phage genome. Firstly, as the mobile genetic element, phages can
mediate horizontal gene transfer between prokaryotes through transduction (Frost et al., 2005).
Hence, the horizontal transfer elements carried in phage genomes are mainly derived from the
prokaryotes they infect, that is, their host. Phages also have a quite small possibility of indirectly
obtaining the genome elements of other phages through the same host. Secondly, as phages must
survive together with their host and are under strong selection pressures during the phage/host co-
evolution, they will adapt the sequence signatures to their host (Edwards et al., 2016). Last but not
least, HoPhage was designed at the beginning based on the cornerstone that sequence signatures
between phages and their host or between phages that infect the same host are similar. Therefore,
the horizontally transferred elements from the host contained in the phage genome will provide
more accurate information for host prediction, and the horizontal transfer element derived from the
phage with the same host will not affect the prediction of the host due to the similar sequence
signatures. The second aspect is about the horizontally transferred elements on the prokaryotic
genome. Horizontal gene transfer (HGT) is a widely recognized mechanism between prokaryotes
and is a major factor in the evolution of prokaryotes (Soucy et al., 2015). The average proportion of
horizontally transferred genes per prokaryotic genome was ~12% of all ORFs, ranging from 0.5%
to 25% depending on the prokaryotic lineage (Nakamura et al., 2004). Although the proportion of
horizontally transferred genes in some prokaryotic genomes is not very low, horizontally transferred
genes will undergo post-HGT compensatory evolution in the recipient genome (Bedhomme et al.,
2019). This is mainly because the transferred genes can differ from the recipient genome in codon
usage preferences, leading to impaired translation and reduced functionality. The post-HGT
compensatory evolution can increase the quantity of the protein expressed from the transferred gene.
In addition, as for exogenous DNA, such as plasmids, the evolution of adaptation to their host is
very rapid through adaptive SNPs (Hulter et al., 2017). The more important thing is that HGT mainly
occurs between closely related organisms and the genetic similarity of the host and the recipient
genomes is high (Soucy et al., 2015). As HoPhage predicts host at the genus level, the transferred
elements are unlikely to cause significant changes in sequence signatures at the genus level. Based
on the above reasons, we think that the impact of the horizontally transferred elements on
prokaryotes on the application of HoPhage is really limited. Based on the above discussion in two
aspects, we believe that the horizontally transferred elements have a limited impact on the usage of
HoPhage for host identification of phages.

In conclusion, HoPhage demonstrates much better performance on short fragments within a
much wider candidate host range. We expect HoPhage to play a vital role in identifying hosts of
novel phages and help researchers to explore the underlying ecological impact of phages in a
community.

Availability of supporting data and materials
The artificial contigs, related scripts, and original results are available at http://cqb.pku.edu.cn/ZhuLab/HoPhage/data/. All the other data are available at corresponding references mentioned in the main text.

HoPhage is user-friendly and does not have high hardware requirements. We have released the
program as a Docker image (https://hub.docker.com/repository/docker/jietan95/hophage) so that
non-computer professionals can use HoPhage without installing any dependency package. Besides, the physical host version of HoPhage can speed up with GPU and is more suitable to handle large-scale data. The program is freely available at http://cqb.pku.edu.cn/ZhuLab/HoPhage/ or https://github.com/jie-tan/HoPhage/.

Reference

Ahlgren, N.A. et al. (2017) Alignment-free $d_2^*$ oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. *Nucleic Acids Res.*, 45, 39-53.

Amgarten, D. et al. (2018) MARVEL, a Tool for Prediction of Bacteriophage Sequences in MetagenomicBins. *Front. Genet.*, 9, 304.

Baker, T.A. (1995) Bacteriophage Mu: a transposing phage that integrates like retroviruses. *Semin. Virol.*, 6, 53-63.

Bedhomme, S. et al. (2019) Evolutionary Changes after Translational Challenges Imposed by Horizontal Gene Transfer. *Genome Biol. Evol.*, 11, 814-831.

Benson, G. (1999) Tandem repeats finder: a program to analyze DNA sequences. *Nucleic Acids Research*, 27, 573-580.

Breitbart, M. and Rohwer, F. (2005) Here a virus, there a virus, everywhere the same virus? *FEMS Microbiol. Rev.*, 29, 39-56.

Carbone, A. (2008) Codon Bias is a Major Factor Explaining Phage Evolution in Translationally Biased Hosts. *J. Mol. Evol.*, 66, 210-223.

Cho, E.H. et al. (2002) Interactions between Integrase and Excisionase in the Phage Lambda Excisive Nucleoprotein Complex. *J. Bacteriol.*, 184, 5200-5203.

Davis, J. and Goadrich, M. The relationship between Precision-Recall and ROC curves. In, *Proceedings of the 23rd international conference on Machine learning. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery*; 2006. p. 233–240.

de Jonge, P.A. et al. (2019) Molecular and Evolutionary Determinants of Bacteriophage Host Range. *Trends Microbiol.*, 27, 51-63.

Edwards, R.A. et al. (2016) Computational approaches to predict bacteriophage-host relationships. *FEMS Microbiol. Rev.*, 40, 258-272.

Fang, Z. et al. (2019) PPR-Meta: a tool for identifying phages and plasmids from metagenomic fragments using deep learning. *GigaScience*, 8.

Fawcett, T. (2006) An introduction to ROC analysis. *Pattern Recognition Letters*, 27, 861-874.

Frost, L.S. et al. (2005) Mobile genetic elements: the agents of open source evolution. *Nat. Rev. Microbiol.*, 3, 722-732.

Galiez, C. et al. (2017) WISH: who is the host? Predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics*, 33, 3113-3114.

Hashemifar, S. et al. (2018) Predicting protein-protein interactions through sequence-based deep learning. *Bioinformatics*, 34, 802-810.

Hayes, S. et al. (2017) Metagenomic Approaches to Assess Bacteriophages in Various Environmental Niches. *Viruses*, 9.

Hulter, N. et al. (2017) An evolutionary perspective on plasmid lifestyle modes. *Curr. Opin. Microbiol.*, 38, 74-80.

Hyatt, D. et al. (2012) Gene and translation initiation site prediction in metagenomic sequences. *Bioinformatics*, 28, 2223-2230.

Koskella, B. and Meaden, S. (2013) Understanding bacteriophage specificity in natural microbial communities. *Viruses*, 5, 806-823.

Li, J. et al. (2014) An integrated catalog of reference genes in the human gut microbiome. *Nat. Biotechnol.*, 32, 834-841.
Lu, C. et al. (2021) Prokaryotic virus host predictor: a Gaussian model for host prediction of prokaryotic viruses in metagenomics. *BMC Biol.*, 19,5.

Mihara, T. et al. (2016) Linking Virus Genomes with Host Taxonomy. *Viruses*, 8,66.

Mock, F. et al. (2020) VIDHOOPHAGE, viral host prediction with Deep Learning. *Bioinformatics*.

Nakamura, Y. et al. (2004) Biased biological functions of horizontally transferred genes in prokaryotic genomes. *Nat Genet.*, 36,760-766.

Pride, D.T. et al. (2006) Evidence of host-virus co-evolution in tetranucleotide usage patterns of bacteriophages and eukaryotic viruses. *BMC Genomics*, 7,8.

Ren, J. et al. (2017) VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. *Microbiome*, 5,69.

Ren, J. et al. (2020) Identifying viruses from metagenomic data using deep learning. *Quant. Biol.*, 8,64-77.

Richter, D.C. et al. (2008) MetaSim—A Sequencing Simulator for Genomics and Metagenomics. *PLoS One*, 3,e3373.

Roux, S. et al. (2015) VirSorter: mining viral signal from microbial genomic data. *PeerJ*, 3,e985.

Roux, S. et al. (2016) Towards quantitative viromics for both double-stranded and single-stranded DNA viruses. *PeerJ*, 4.

Shkoporov, A.N. and Hill, C. (2019) Bacteriophages of the Human Gut: The "Known Unknown" of the Microbiome. *Cell Host Microbe*, 25,195-209.

Smits, S.L. et al. (2014) Assembly of viral genomes from metagenomes. *Front. Microbiol.*, 5,714.

Soucy, S.M. et al. (2015) Horizontal gene transfer: building the web of life. *Nat. Rev. Genet.*, 16,472-482.

Suzuki, H. et al. (2010) Predicting Plasmid Promiscuity Based on Genomic Signature. *J. Bacteriol.*, 192,6045-6055.

Szegedy, C. et al. Rethinking the Inception Architecture for Computer Vision. In, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016. p. 2818-2826.

Ugorcakova, J. and Bukovska, G. (2003) Lysins and holins: tools of phage-induced lysis. *Biologia*, 58,327-334.

Wang, W. et al. (2020) A network-based integrated framework for predicting virus-prokaryote interactions. *NAR: Genomics Bioinf.*, 2,1qaa044.

Yang, C. et al. (2018) LncADeep: an ab initio lncRNA identification and functional annotation tool based on deep learning. *Bioinformatics*, 34,3825-3834.

Yuan, Y. and Bar-Joseph, Z. (2019) Deep learning for inferring gene relationships from single-cell expression data. *Proc. Natl. Acad. Sci. U. S. A.*, 116,27151.

Zhang, M. et al. (2017) Prediction of virus-host infectious association by supervised learning methods. *BMC Bioinf.*, 18,60.

Zhao, X. et al. (2012) Coevolution between simple sequence repeats (SSRs) and virus genome size. *BMC Genomics*, 13,435.