OXFORD UNIVERSITY PRESS | Briefings In Bioinformatics

CHERRY: a Computational metHod for accuratE pRediction of virus-pRokarYotic interactions using a graph encoder-decoder model

Journal:	*Briefings in Bioinformatics*
Manuscript ID	BIB-22-0013
Manuscript Type:	Problem solving protocol
Date Submitted by the Author:	03-Jan-2022
Complete List of Authors:	Shang, Jiayu; City University of Hong Kong Sun, Yanni; City University of Hong Kong
Keywords:	Phage host prediction, Link prediction, Graph convolution network, Deep learning

Journal Title Here, 2019, 1–14
doi: DOI HERE
Advance Access Publication Date: Day Month Year
Paper

CHERRY: a Computational metHod for accuratE pRediction of virus-pRokarYotic interactions using a graph encoder-decoder model

Jiayu Shang and Yanni Sun*

Department of Electrical Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, China SAR

*Corresponding author. yannisun@cityu.edu.hk

FOR PUBLISHER ONLY Received on Date Month Year; revised on Date Month Year; accepted on Date Month Year

Abstract

Prokaryotic viruses, which infect bacteria and archaea, are key players in microbial communities. Predicting the hosts of prokaryotic viruses helps decipher the dynamic relationship between microbes. Although there are experimental methods for host identification, they are either labor-intensive or require the cultivation of the host cells, creating a need for computational host prediction. Despite some promising results, computational host prediction remains a challenge because of the limited known interactions and the sheer amount of sequenced phages by high-throughput sequencing technologies. The state-of-the-art methods can only achieve 43% accuracy at the species level. This work presents CHERRY, a tool formulating host prediction as link prediction in a knowledge graph. As a virus-prokaryotic interaction prediction tool, CHERRY can be applied to predict hosts for newly discovered viruses and also the viruses infecting antibiotic-resistant bacteria. We demonstrated the utility of CHERRY for both applications and compared its performance with the state-of-the-art methods in different scenarios. To our best knowledge, CHERRY has the highest accuracy in identifying virus-prokaryote interactions. It outperforms all the existing methods at the species level with an accuracy increase of 37%. In addition, CHERRY's performance is more stable on short contigs than other tools.

Key words: Phage host prediction, Link prediction, Graph convolution network, Deep learning

Key Messages

- In this work, we present CHERRY, a new virus-prokaryote interaction prediction tool. Our large-scale experiment on finding the interactions for 1,940 viruses and 60,105 prokaryotic genomes shows that CHERRY improves the host prediction accuracy from 43% to 80% at the species level.
- Our rigorous test of CHEERY on other datasets and the benchmark experiments against 8 recently published tools show that CHERRY is the most accurate host prediction tool.
- Unlike existing tools, CHERRY can be flexibly used in two scenarios. It can take either query viruses or prokaryotes as inputs. For input viruses, it can predict their hosts. For input prokaryotes, such as antibiotic-resistant bacteria, it can predict the viruses infecting them.

Introduction

Prokaryotic viruses (shortened as viruses hereafter), including bacteriophages and archaeal viruses, are highly ubiquitous and abundant. They are key players in a wide range of microbial communities. By infecting their hosts, they can regulate both the composition and function of the microbiome. Thus, identifying the hosts of novel viruses play an essential role in characterizing the interactions of the organisms inhabiting the same niche. In addition, because phages can kill the bacterial hosts through lytic infections, making them a promising

therapeutic strategy for treating bacterial infections. The corresponding application named phage therapy [16] received increasing attention because of the fast rise of antibiotic-resistant pathogens. Many experiments have demonstrated that phage therapy is a potential alternative to antibiotics for killing the "superbugs" [35, 5, 38]. Applying phage therapy for treating bacterial infections in humans is under active development with dozens of registered clinical trials at NIH of USA.

Despite the importance of the virus-host interactions, the characterized interactions between viruses and prokaryotes is just the tip of the iceberg. Experimental methods for host identification can be labor intensive or require the cultivation of the prokaryotes [23], limiting their large-scale applications. As reported in [25, 49], no more than 1% of microbial hosts can be cultivated successfully in laboratories. With advancements of high-throughput sequencing, a large number of new phages can be sequenced without the need of host cell cultivation. As a result, host identification for newly sequenced phages lagged behind the fast accumulation of the sequencing data. Thus, computational approaches for host prediction are in great demand.

There are two specific challenges for computational host prediction. The first one is the lack of known virus-host interactions. For example, the number of known interactions dated up to 2020 only accounted for ˜40% (1,940) of the prokaryotic viruses at the NCBI RefSeq at that time. Meanwhile, among the 60,105 prokaryotic genomes at the NCBI RefSeq, only 223 of them have annotated interactions with the 1,940 viruses. The limited known interactions may not provide enough information for host prediction. Second, although sequence similarity between viruses and hosts is an insightful feature for host prediction, not all viruses share common regions with their host genomes. For example, in the RefSeq database, ˜24% viruses do not have significant alignments with their hosts. Therefore, the alignment-based methods cannot identify hosts for some phages.

Related work

Several computational methods were developed to predict hosts for viruses [18, 24, 26, 2, 47]. According to the design, these methods can be roughly divided into two groups: alignment-based and learning-based. The alignment-based methods utilize the similarity search between either the reference viruses and query viruses or the reference prokaryotes and query viruses for host identification. For example, VPF-Class [39] utilizes Viral Protein Families (VPFs) downloaded from the IMG/VR system to estimate the similarity between query viruses and viruses with known hosts. According to the alignment results with VPFs, VPF-class can return predictions for each query contig. Some other methods use marker genes for host prediction. For example, when infecting the host cell, the receptor-binding protein (RBP) of the virus helps inject the virus genome into the host [10]. Thus, RBP-based similarity can be used for host prediction. However, it is not trivial to annotate RBPs in all viruses. Alignment-based tools also use CRISPR for host prediction. Some prokaryotes can keep a record of virus infection via CRISPR [24], whose spacers contain short nucleotide sequences from infected viruses for preventing recurring infection [1]. Thus, local alignment programs such as BLAST [33] can be employed to predict hosts by searching for short matches between prokaryotes and viruses. However, it is estimated that only ˜20% of sequenced prokaryotes contain CRISPRs [13, 30]. Although CRISPR is an informative host prediction signal, many viruses do not have alignment results with the annotated or predicted CRISPRs of prokaryotes and will return no predictions.

Learning-based methods are more flexible. Most of these methods learn sequence-based features for host prediction. For example, WIsH [26] trains a homogeneous Markov model for potential host genomes. The model will then calculate the likelihood of a prokaryote genome as the host for a query virus and assigns the host with the highest likelihood. vHULK [4] formulates host prediction as a multi-class classification problem where the inputs are viruses and the labels are the prokaryotes. The features used in their deep learning model is the protein profile alignment results against pVOGs database of phage protein families [29]. Rather than using the public database, RaFAH [19] uses MMseqs2 [44] to generate protein clusters and construct profile hidden Markov models (HMMs). Then, they use features output by the HMM alignments and train a multi-class random forest model. HoPhage [45], another multi-class classification model-based host prediction tool, uses deep learning and Markov chain algorithm. They use the CDS of each candidate host genome to construct a Markov chain model and then calculate the likelihood of query phage fragments infecting the candidate host genomes. They also use a deep learning model and finally integrate the results of deep learning model with the Markov model for host prediction. On the other hand, PHP [36] utilizes the k-mer frequency, which can reflect the codon usage patterns shared by the viruses and the hosts [28, 15]. HostG [43] utilizes the shared protein clusters between viruses and prokaryotes to create a knowledge graph and trains a graph convolutional network for prediction. However, most of these tools can only predict the host at the genus level. The best host prediction performance at the species level is reported by VHM-Net [47], which incorporates multiple features between viruses and prokaryotes such as CRISPRs, the output score of WIsH, BLASTN alignments, etc. By combining these features, VHM-net utilizes the Markov random field framework for predicting whether a virus infects a target prokaryote. Nevertheless, the accuracy at the species level is only 43% and has a large room to improve.

Overview

In this work, we develop a new method, CHERRY, which can predict the hosts' taxa (phylum to species) for newly identified viruses. First, we construct a multimodal graph that incorporates multiple types of interactions, including protein organization information between viruses, the sequence similarity between viruses and prokaryotes, and the CRISPR signals (Fig. 1 A). In addition, we use k-mer frequency as the node features to enhance the learning ability. Second, rather than directly using these features for prediction, we designed an encoder-decoder structure to learn the best embedding for input sequences and predict the interactions between viruses and prokaryotes. The graph convolutional encoder (Fig. 1 B) will utilize the topological structure of the multimodal graph and thus, features from both training and testing sequences can be incorporated to embed new node features. Then a link prediction decoder (Fig. 1 C) is adopted to estimate how likely

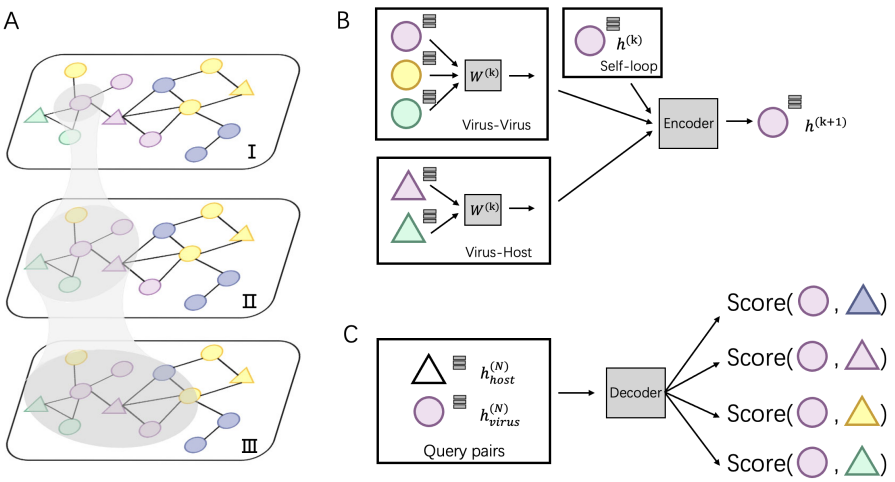

Fig. 1: The key components of CHERRY. A) The multimodal knowledge graph. Triangle represents the prokaryotic node and circle represents virus nodes. Different colors represents different taxonomic labels of the prokaryotes. I-III illustrate graph convolution using neighbors of increasing orders. B) The graph convolutional encoder of CHERRY. C) The decoder of CHERRY.

a given virus-prokaryote pair forms a real infection. Unlike existing tools, CHERRY can be flexibly used in two scenarios. It can take either query viruses or prokaryotes as input. For viruses, it can predict their hosts. For prokaryotes of interest, mainly the pathogenic bacteria, it can predict the viruses infecting them. Another feature behind the high accuracy of CHERRY is the construction of the negative training set. The dataset for training is highly imbalanced, with the real host as the positive data and all other prokaryotes as negative data. We carefully addressed this issue using negative sampling [37]. Instead of use a random subset of the negative set for training the model, , we apply end-to-end optimization and negative sampling to automatically learn the hard cases during training. To demonstrate the reliability of our method, we rigorously tested CHERRY on multiple datasets including the RefSeq dataset, simulated short contig dataset, and metagenomic dataset. We compared CHERRY with WIsH, PHP, HoPhage, VPF-Class, RaFAH, HostG, vHULK, and VHM-net; the results show that CHERRY competes favorably against the state-of-the-art tools and yields 37% improvements at the species level.

Method

We formulate host prediction as a link prediction problem [3] on a multimodal graph, which encodes virus-virus and virus-prokaryote relationships. To be specific, these relationships can be represented by a knowledge graph $G = (V, E)$ with node $v_i \in V$, where $i = 1, 2, ..., N$. An edge between v_i and v_j is denoted as a tuple $(v_i, v_j) \in E$. There are two kinds of nodes in the graph: viral nodes $p_i \in P$ and prokaryotic nodes $h_i \in H$ ($P \cup H = V$). Then the link prediction task can be defined as: **given** a viral node p_i and a prokaryotic node h_i, **what is the probability** of p_i and h_i having a link (infection). In the following section, first, we will describe how we construct the multimodal graph. Then, we will introduce the encoder-decode structure of CHERRY.

Construction of the knowledge graph G

To utilize the features from both training and testing samples, we construct a multimodal graph G by connecting viruses and prokaryotes in both the reference database and the test set. This multimodal graph is composed of protein organizations, sequence similarity, and CRISPR-based similarity. The node in the graph encodes k-mer frequency feature from the DNA sequences. According to the type of the connections, the edges in the knowledge graph can be divided into virus-virus connections and virus-prokaryote connections.

Virus-virus connections: we utilize the protein organizations to measure the similarity of biological functions between viruses. Intuitively, if two viruses share similar protein organizations, they are more likely to infect the same host. First, we construct protein clusters using the Markov clustering algorithm (MCL) on all viral proteins. For reference viral genomes, the proteins are downloaded from NCBI RefSeq. For query contigs, we use Prodigal [32] to conduct gene finding and protein translation. Then, we employ MCL to cluster proteins with *inflation* = 2.0 based on the DIAMOND BLASTP [12] comparisons (E-value <1e-5). Second, we followed [11, 42] and use Eq. 1 to to estimate the probability of two viruses X and Y sharing at least c common protein clusters by assuming that each protein cluster has the same chance to be chosen. x and y are the numbers of proteins in X and Y, respectively. Because Eq. 1 computes the background probability under the hypothesis that virus X and Y don't share common host, we will reject this hypothesis when P is smaller than a cutoff. Finally, only pairs with P($\geq$ c) smaller than τ_1 will form virus-virus connections (Eq. 2).

$$P(\geq c) = \sum_{i=c}^{min(x,y)} \frac{\binom{x}{i}\binom{n-x}{y-i}}{\binom{n}{y}} \quad (1)$$

$$virus\text{-}virus = \begin{cases} 1 & \text{if } P(\geq c) < \tau_1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Virus-prokaryote connections: we applied the sequence similarity between viral and prokaryotic sequences to define the virus-prokaryote connections. There are two kinds of sequence similarity that can be employed: CRISPR-based and general local similarity. Some prokaryotes will integrate some viral DNA fragments into their own genomes to form spacers [22, 40] in CRISPR. Therefore, many existing tools have used CRISPR as a main feature for host prediction [24, 48]. In our method, CRISPR Recognition Tool [9] is applied to capture potential CRISPRs from prokaryotes. If a viral sequence shares a similar region with the CRISPR, we will connect this viral node to the prokaryotic node.

However, only a limited number of CRISPRs can be found. We thus also use BLASTN to measure the sequence similarity between the sequences. Viruses can mobilize host genetic material and incorporate it into their own genomes. Occasionally, these genes can bring an evolutionary advantage and the viruses will preserve them [24]. For all viruses $p_i \in P$ and prokaryotes $h_i \in H$, we will run BLASTN for each pair (p_i, h_i). Only pairs with BLASTN E-value smaller than τ_2 will form virus-prokaryote connections. In addition, because we have known virus-prokaryote connections from the public dataset, we connect the viruses with their known hosts regardless of their alignment E-values. Finally, the edges $(p_i, h_i) \in E$ can be formulated as Eq. 3. If there is an overlap between CRISPR-based and BLASTN-based edge, we will only create one edge between the virus and prokaryote.

$$virus\text{-}prokaryote = \begin{cases} & \text{if } \exists \text{ CRISPR alignment} \\ 1 & \text{or BLASTN } E_{value} < \tau_2 \\ & \text{or } \exists \text{ interaction in dataset} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Both τ_1 and τ_2 are the default parameters given in [42] and [14].

Nodes construction: The nodes in the multimodal graph represent the features from both viral and host genomes. In our work, we utilize the k-mer frequency as our node feature. Since the dimension of the feature vector increases exponentially with k, we choose $k = 4$ and obtain a 256-dimensional vector for each viral or host genome.

The encoder-decoder framework in CHERRY

After constructing the multimodal graph G, we will feed it to our model for training and prediction. Given a virus-prokaryote pair (p_i, h_i), our aim is to determine how likely there is a link (infection) between virus p_i and prokaryote h_i. Towards this goal, we develop a non-linear, multi-layer decoder-encoder network CHERRY that operates on the multimodal graph G. CHERRY has two main components:

- *Graph convolutional encoder:* a graph convolutional network operating on G and embedded node features in G (Fig. 1 B)
- *Query pairs decoder:* a 2-layer neural network classifier that can output a probability score for each virus-prokaryote pair (Fig. 1 C)

Detailed information about the two components and the end-to-end training method will be discussed in the following sections.

Graph convolutional encoder

The input of the encoder is the multimodal graph G. Since we have both labelled (viruses with known hosts) and unlabelled (viruses without known hosts) nodes in the graph, the main idea of the encoder is to utilize the topological structure to propagate information from labelled nodes to unlabelled nodes. The output of the encoder will be d-dimensional embedded vectors that integrate virus-virus similarity and virus-prokaryote similarity gained from the multimodal graph G.

We will use graph convolutional neural network (GCN) to conduct feature embedding of the nodes. GCN is one well-studied model for graph data. Recently, it has some successful applications in biological data [42, 20]. In our encoder, we will take advantage of the feature embedding of GCN to encode the feature vectors for viruses and prokaryotes. Specifically, for a given node (Fig. 1 AI), our encoder performs convolution on its neighbors' node vectors and itself. In each convolution operation, the encoder considers the 1-step neighborhood of the nodes (Fig. 1 AII) and applies the same transformation to all nodes in the graph. Then the successive convolution will be applied in the l layers, and finally, each node will effectively convolve the information from its l-step neighborhood (Fig. 1 AIII). l is the number of the graph convolutional layer in the encoder. A single graph convolutional layer can be represented as Eq. 4

$$h^{l+1} = \phi(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}h^{(l)}W^{(l)}) \quad (4)$$

ϕ is the activation function in the graph convolutional layer. $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix, where N is the number of nodes in the multimodal graph. $\tilde{A} = A + I$, where $I \in \mathbb{R}^{N\times N}$ is the identity matrix. $\tilde{D}$ is the diagonal matrix calculated by $D_{ii} = \sum_j \tilde{A}_{ij}$. $h^{(l)}$ is the hidden feature in the lth layer and $h^{(0)} \in \mathbb{R}^{N\times 256}$ is the 4-mer frequency vector of each node. W^l is a matrix of the trainable filter parameters in the l-layer. Finally, the encoder will output a d-dimensional embedded vector, which encoded prior knowledge from l-step neighborhood for each node in the graph. d is the dimension of W in the output layer. Because the convolutional layer will be conducted on all nodes in G, features from both training and testing samples can be used in the encoder to enhance the learning ability.

Decoder for link prediction

After encoding the feature vectors for viral and host genomes, we apply a 2-layer neural network classifier to decode the embedded vectors outputted from the encoder. This decoder aims to judge how likely these query pairs form actual infections. Thus, The input of the decoder is a query set Q, and the output of the decoder is a probability score. Each element in Q is called a query vector q_{ij} and is calculated by Eq. 5.

$$q_{ij} = encoder(p_i) - encoder(h_j) \quad (5)$$

First, we generate all-against-all virus-prokaryote pairs and calculate all query vectors $q_{ij} \in Q$. $encoder(\cdot)$ represent the output of graph convolutional encoder. $p_i \in P$ represents the virus node, and $h_j \in H$ represents the prokaryotic node. Then we employ a 2-layer neural network to decode the feature vector for each input q_{ij} as shown in Eq. 6.

$$\begin{cases} q_{ij}^{(l+1)} = \phi(q_{ij}^{(l)}\theta^{(l)}) \\ decoder(q_{ij}) = sigmoid(q_{ij}^{(L-1)}) \end{cases} \quad (6)$$

$q_{ij}^{(l)}$ is the hidden feature in the lth layer. $q_{ij}^{(0)} = q_{ij}$ and L is the maximum number of the layers in the network. ϕ is the activation function. $decoder(\cdot)$ represent the output of the link prediction decoder. Because the activation function of the output layer is the *sigmoid* function, $decoder(\cdot)$ can be used as the probability score for each pair.

Model training

Recent results show that the modeling of topological structure data can be greatly improved through end-to-end learning [21, 27]. Thus, we optimize overall trainable parameters for CHERRY and backpropagate loss on the multimodal graph. The trainable parameters of CHERRY are: (i) convolutional filters' weight matrices W in the encoder and (ii) query parameter matrices θ in the decoder.

$$J(i,j) = -logP_r^{ij} - \mathbb{E}_{n\sim P_r} log(1 - P_r^{ik}) \quad (7)$$

There are two kinds of query pairs that will be generated by Eq. 5: positive pairs and negative pairs. Positive pairs represent known virus-prokaryote interactions given by the dataset. Negative pairs represent the pairs with no evidence for interaction. Then, during the training process, we optimize the model using the cross-entropy loss as shown in Eq. 7. This equation encourages the model to assign higher probabilities to positive pairs (p_i, h_j) than the randomly created negative pair (p_i, h_k).

Because we form all-against-all query pairs from all viruses and hosts, the number of negative query pairs will be much larger than the positive query pairs. To solve this problem, rather than sampling a subset of the negative pairs, we optimize the model through negative sampling, a method introduced in recent publications [37, 46]. When calculating the loss in training, we replace prokaryotic node h_j in positive pair (p_i, h_j) with prokaryotic node h_k that is selected according to a sampling distribution P_r defined in [37]. Intuitively, if a negative sample has a higher loss, it will have a higher probability to be selected for training. Thus, the negative sampling method helps enhance the learning ability of the model compared to sub-sampling because the latter cannot represent the real distribution of the sample space, especially when the number of negative samples is much more than the number of positive samples. With negative sampling, CHERRY can automatically learn the hard cases using the negative sampling algorithm.

To ensure that the optimization process can learn as many query pairs as possible during the negative sampling process, we train it for a maximum of 250 epochs using the Adam optimizer [34] with a 0.01 learning rate to update the parameters. Before evaluating our model, we fixed the random seed in the program to ensure that we had the same initial parameters. To guarantee the model's reliability, we also save the parameters so that users can directly load the parameters when applying CHERRY on their own dataset. Users can also use the parameters as a pre-trained model and add more interactions for training, which will help the model converge faster.

Experimental setup

This section will introduce how we evaluate our model and compare it with the state-of-the-art tools.

Metrics: according to the usage of link prediction, CHERRY can be employed in two different scenarios: 1. Predicting host for newly identified viruses; and 2. Identifying viruses that infect targeted pathogenic bacteria. Thus, we apply two different evaluation methods, respectively.

Predicting hosts for newly identified viruses

We use the same experimental setup and metrics as the previous works to ensure consistency and a fair comparison with the state-of-the-art tools. Following the previous work, one virus is assumed to infect only one host, which is not always true but is a commonly used assumption for evaluation. Thus, in the experiments, for each testing virus p_i, we will compute the score between p_i and all prokaryotes in set H and output the prokaryote with the highest decoder score (Eq. 8).

$$\arg\max_{h_j \in H} \ decoder(q_{ij}) \quad (8)$$

p_i is a testing virus, and H is the set of prokaryotic genomes. q_{ij} is defined in Eq. 5. We predict hosts for all viruses in the test set and evaluate the performance by judging whether the predicted prokaryotes are from the same taxon (such as same species) as the given interactions in the dataset. The ratio of correct prediction is the *accuracy*, which has the same definition as previous works. In addition, because the graph contains all the potential hosts (prokaryotes) and the number of predictions is equal to the number of test viruses, the recall and precision are the same as the accuracy. It is worth noting that some benchmark tools use *recall* to represent the number of viruses with predictions. To avoid conflicts, we call this metric prediction rate.

Identifying viruses that infect pathogenic bacteria

The goal is to identify which viruses can infect a specific prokaryote. Because other methods do not support this utility, we define our own evaluation metrics. In this scenario, one prokaryote can be infected by multiple viruses. We will set a threshold μ to decide the set of predicted viruses.

$$S(h_j) = \{\forall p_i \in P \mid \ if \ decoder(q_{ij}) > \mu\} \quad (9)$$

As shown in Eq. 9, for each prokaryote, we will predict a set of viruses whose probabilities (calculated by the decoder) are larger than μ. Unlike the host prediction task, this equation might predict viruses infecting different prokaryotes as long as the probability is larger than the threshold. Then, for each prokaryote h_j, the precision represents how many viruses in $S(h_j)$ truly infect h_j based on the ground truth. The recall represents how many viruses infecting h_j are in $S(h_j)$.

Dataset: we followed [36] and used the same virus-host relationship benchmark dataset for training (the VHM dataset) and testing (the TEST dataset). The detailed information is shown in Fig. 2 (A). We download all 1,940 viruses from the NCBI RefSeq database and separate the training set and test set according to their submission time (before 2020). Finally, we have 1,306 positive pairs for training and 634 positive pairs for testing. Although every virus is unique, some of them infect the

same host. Training set and test set share 59 host species. To show the overall similarity between the training set and test set, we use Dashing [6] to estimate the sequence similarity between phages. We record the largest value for each testing phage against all training phages and report the overall similarity in Fig. 2 (B). The result reveals that only a few testing phages are similar to the training phages, with the mean similarity value being 0.1. Along with 233 host species, we also have 60,105 prokaryotic genomes obtained from the NCBI genome database before 2020.

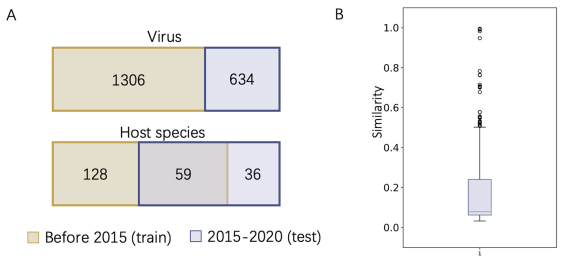

Fig. 2: Virus-host interaction dataset. (A) Dataset used for training and testing. (B) Similarity between viruses used in testing and training sets.

To further assess the utility of CHERRY on predicting hosts for newly discovered viruses, we applied it to viruses from two recently published metagenomic datasets. In addition, we employed CHERRY to search for phages that can infect antibiotic-resistant bacteria. This application can provide new knowledge for constructing the phage cocktail in phage therapy. The information of the datasets are summarized as below:

- *Glacier metagenomic dataset*: This is a newly published metagenomic dataset sampled from the ice core on the Tibetan Plateau [50]. The dataset presents 33 new phage contigs and four dominant bacteria genus isolated from the sample. The dataset is available from: `https://datacommons.cyverse.org/browse/iplant/home/shared/iVirus/Tibet_Glacier_viromes_2017`.
- *Gut metagenomic dataset*: 3,738 previously unknown phages were discovered in human gut metagenomic data [8]. These phage genomes represent 451 putative genera whose hosts remain unknown. The dataset is available from: `ftp://ftp.ncbi.nih.gov/pub/yutinn/benler_2020/gut_phages/`.
- *Predicting phages that infect antibiotic-resistant bacteria*: This dataset has three antibiotic-resistant bacteria (superbugs) that cause serious health problems but are resistant to antibiotic treatment. Recent research shows evidence that some phages can help kill these bacteria. We will evaluate whether CHERRY can identify the interactions between phages and these superbugs. The accession numbers of these phages and superbugs can be found in the our GitHub folder.

Result

In this section, we will show our experimental results on different datasets and compare CHERRY against the state-of-the-art tools: WIsH [26], PHP [36], VHM-Net [47], VPF-Class [39], vHULK [4], RaFAH [19], and HostG [43]. We also recorded the host prediction performance using either BLASTN or CRISPRs, which are two frequently used features for identifying the interactions. The experimental results consist of two parts.

Predicting hosts for viruses

First, we will report the host prediction performance across different taxonomic levels (from species to Phylum) on the benchmark dataset. We will analyze how the knowledge graph and the encoder-decoder structure improve the host prediction accuracy. In addition, we will investigate why the alignment-based method fails to perform well in the host prediction task. Second, we will show the host prediction results on the short contigs to evaluate the robustness of CHERRY. Third, we will visualize the results of host prediction on different viral families and analyze why prediction at the species level is a hard case for the host prediction. Then, we present a further improvement of CHERRY to narrow the search scope of the potential host for newly identified viruses. Finally, we will show two case studies to demonstrate the reliability of CHERRY on metagenomic data.

Predicting viruses that infect pathogenic bacteria

First, we will present the precision and recall of virus identification for given hosts. Then, we will show a case study in which we use CHERRY to predict phages that can infect antibiotic-resistant bacterial pathogens.

Visualization of the knowledge graph

We use Gephi [7] to visualize the knowledge graph in Fig. 3. We colored the nodes by their labels. For the prokaryotic node, the label is the species taxonomy. For a virus node, the label represents its host species. Because there are too many labels/species in the graph, we only colored eight labels/species with the largest proportions. All other nodes will be in grey.

Fig. 3 shows the topological structure of the knowledge graph. The GCN encoder will utilize the edge information of the graph to embed the node features. If the edges only connect nodes with the same label, a simple label propagation method based on graph connectivity can achieve accurate prediction. However, although there are some clusters containing nodes with pure labels (e.g. region A and region B), some subgraphs are mixed with multiple labels. For example, region C contains more than six labels. Nodes (viruses or prokaryotes) in such a region often connect to nodes with different labels. In addition, we also found that prokaryotic nodes usually connect to the virus nodes with different labels. For example, node A1 represents *Cutibacterium acnes*, but it has many edges (alignments) with phages that infect *Mycobacterium smegmatis*. In this case, simple alignment-based or label propagation in the graph will fail to make the correct host prediction. But the GCN-based encoder can integrate both the sequence similarity, k-mer composition, and the topological information for making correct predictions in this graph.

Predicting host for prokaryotic viruses

In this experiment, we used the dataset shown in Fig. 2 for training and evaluation. For each test virus, our predicted host is the one with the highest prediction score out of the 60,105 prokaryotes. Then we calculate the accuracy according to the given host in the dataset.

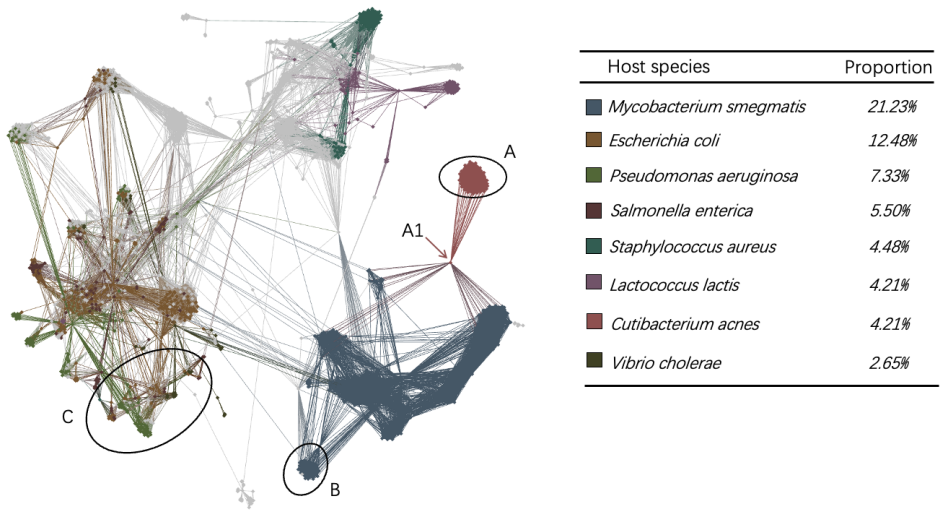

Fig. 3: Visualization of the multimodal graph. The colors of the nodes are their labels: for the prokaryotic node, the labels represent the species. For the viral node, the label represents the host species. Because there are a large number of labels, this graph only colors the top 8 labels with the largest number of nodes. All others are gray.

Model training

To train a reliable model and evaluate the overall learning ability of CHERRY, we applied 10-fold cross-validation on the training set. First, we randomly split the training set into ten subsets. Then, we trained the model on nine subsets, validated it on the tenth iteratively, and recorded the prediction performance. As shown in Fig. 4(A), we reported the highest, lowest, and average performance of the 10-fold cross-validation from species to phylum level. We also reported the BLAST results of the host prediction. We used 60,105 prokaryotes as reference genomes and then recorded the alignment results. Then we predict the host using k nearest neighbor with $k = 1$ (the best alignment) or > 1 (majority vote). Here, we only reported the best alignment method because it has higher accuracy. Fig. 4(A) shows that CHERRY largely improves the performance at the species level. Also, it is worth noting that only ˜65.5% and ˜24.6% viruses are predicted by BLASTN and CRISPRs, respectively. However, CHERRY can predict labels for all viruses.

In addition, we investigated how different components in the knowledge graph affect the prediction accuracy. Our knowledge graph is constructed by two types of edges: virus-virus and virus-prokaryote. We show how the edges affect the accuracy in Fig. 4(A). We use negative sampling method to train on the four graphs and the definition of each graph is listed below.

- *without graph*: We trained the model without graph (without encoder). The decoder will only use the k-mer frequency vectors as inputs.
- *virus-virus*: We trained the model with a graph that only contains virus-virus edges.
- *virus-prokaryote*: We trained the model with a graph that only contains virus-prokaryote edges.
- *complete graph*: We trained the model with the complete multimodal graph.

The results show that both virus-virus similarity and virus-prokaryote similarity can enhance the learning ability and improve performance. The complete knowledge graph that contains two types of edges achieves the best performance.

We also show the comparison between training with random sampling a subset and negative sampling in Fig. 4(A). The prediction results show that negative sampling can largely improve the host prediction accuracy using negative sampling method.

Evaluation on the test set

After training the model, we used the parameters with the highest validation accuracy to predict the hosts for viruses in the test set. To conduct a fair comparison, we also re-trained the state-of-the-art models to record their results. Because VPF Class is an alignment-based method, we used their database for host prediction. As shown in Fig. 4 (B), CHERRY achieved the best accuracy of 78% in predicting the hosts' species, which is 35% higher than the next best tool VHM-net. To prove that the convolutional graph encoder can enhance the learning ability of CHERRY, we reused the model *without graph* in Fig. 4(A) for host prediction. The final results on the test set decrease to 56%, and thus, the convolutional graph encoder helps host prediction. This experiment also shows that VHM-net, vHULK, HostG, and RaFAH have better performance than other tested tools and thus we will only keep these tools in future experiments.

Training vs testing similarity can impact the accuracy

In the *Dataset* section, we use Dashing to measure the similarity between the test set and training set. To show how the similarity affects the host prediction performance, we divided the test set according to the dashing similarity. We recorded the accuracy at the genus level in Fig. 5. X-axis stands for the maximum similarity between genomes in the training set and test set. For example, when the X-axis value is 0.8, all the genomes in the test set have similarity ≤ 0.8 against the genomes in the training set. Fig. 5 also shows how the similarity influences other four

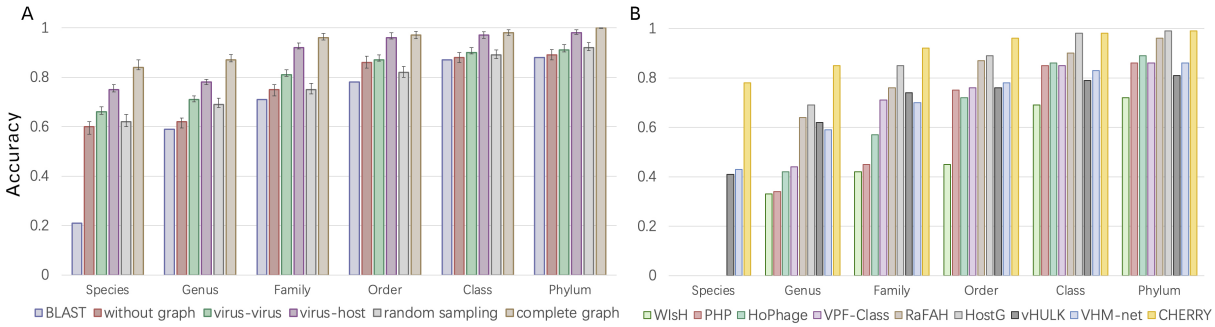

Fig. 4: Host prediction evaluation on the benchmark dataset. Y-axis: accuracy. (A) 10-fold cross-validation on the training set. X-axis represents different types of graphs and BLAST-based host prediction. *without graph*: training with only decoder. *Virus-virus*: the knowledge graph only contains *virus-virus* edges. *virus-prokaryote*: the knowledge graph only contains *virus-prokaryote* edges. *random sampling*: the model is trained on the complete graph with a randomly sampled negative set. *complete graph*: the model is trained on the complete graph with negative sampling. Error bar represents the highest, lowest, and average accuracy of the 10-fold cross-validation. (B) Comparison of host prediction accuracy on the test set from species to phylum level. Some tools cannot output predictions at the species level and thus we can only measure species-level accuracy for vHULK, VHM-net, and CHERRY.

tools that have relatively good performance in the experiment shown in Fig. 4.

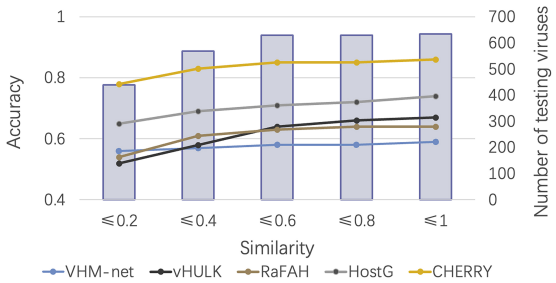

Fig. 5: Associations between accuracy and similarity at the genus level. X-axis: dashing similarity. Left Y-axis: Accuracy. Right Y-axis: number of testing viruses under each similarity cutoff.

As expected, with the increase of the similarity, more testing genomes with high similarities are included, and the accuracy of all methods increases. The gap between CHERRY and all other methods clearly shows that our model outperforms the state-of-the-art tools on a wide range of similarities.

Sequence similarity between viruses and prokaryotes

Then, we further investigated whether the host prediction tools can handle the case of viruses lacking significant similarities with prokaryotic genomes. All the testing viruses that do not have BLASTN alignments with the reference prokaryotes are used for evaluation. As shown in Fig. 6, the accuracy of all methods decreases; but CHERRY still renders the best performance. vHULK, RaFAH, and HostG have better robustness than VHM-net because they mainly rely on the phage protein similarity for host prediction. The experimental results shown in Fig. 5 and Fig. 6 reveal that by integrating the virus-virus relationships and virus-prokaryote relationships, the multimodal graph enhances the learning ability and host prediction performance.

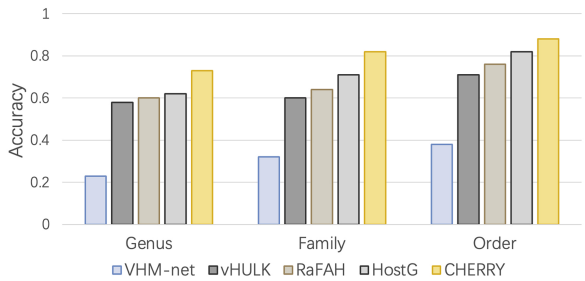

Fig. 6: Host prediction accuracy for contigs without significant alignment results against the prokaryotes. X-axis: taxonomic rankings. Y-axis: accuracy.

Hard cases for alignment-based methods

As stated in Introduction, using the similarity between the viruses and the prokaryotes alone cannot always provide precise host identification. An example is given in Fig. 7, where colored nodes represent training viruses (with host species labels) and nodes in white represent testing viruses (without labels). Different color represents different prokaryotic species/labels. Triangle nodes are prokaryotic nodes, and circle nodes are virus nodes. The testing viruses (white nodes) share high similarities with multiple nodes in different colors. Also, if we use the majority vote methods, node 'Test_120', 'Test_64', and 'Test_178' will be predicted as *Bacillus subtilis*. However, according to the given labels in the database, only 'Test_178' belongs to *Bacillus subtilis* virus. Virus 'Test_120' and 'Test_64' are *Bacillus thuringiensis*.

While these heterogeneous connections pose challenges for other tools, CHERRY is able to predict the hosts correctly. We recorded the prediction score of CHERRY in Table 1. We also reported the prediction results of vHULK and

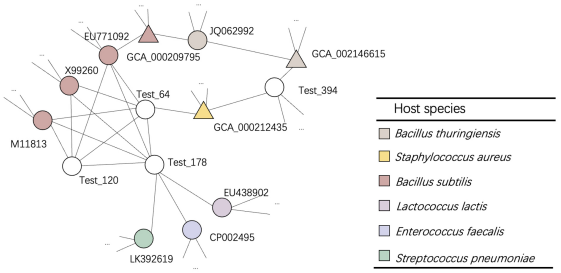

Fig. 7: A case study for alignment-based method. Part of the multimodal graph shows that only using alignment-based method failed to give a correct prediction for 'Test_120' and 'Test_64'. Triangle represents prokaryotic nodes and circle represents virus nodes. Nodes with color represent training samples and different color represents different species label. Nodes in white represent test samples. The open-end edges beside the nodes indicate that these nodes might have more connections.

VHM-net because they can predict hosts at the species level. The results show that CHERRY can predict both viruses correctly and vHULK can assign host to one of the viruses correctly. VHM-net failed to predict both of them. As introduced in Section *Visualization of the knowledge graph*, a plausible explanation is CHERRY not only considers alignment similarity/connections in the multimodal graph but also considers the k-mer frequency of the genomes when training the link prediction decoder. In summary, 73.7% of viruses have multiple alignments/connections with different labelled nodes (viruses or prokaryotes). While alignment-based methods can return ambiguous/wrong predictions, CHERRY can provide more accurate host identification for these viruses.

Method	Virus	*Bacillus thuringiensis*	*Bacillus subtilis*	*Staphylococcus aureus*
CHERRY	Test_120	0.996	0.416	0.003
	Test_64	0.985	0.115	0.002
vHULK	Test_120	-	0.819	-
	Test_64	0.771	-	-
VHM-net	Test_120	-	0.991	-
	Test_64	-	-	0.988

Table 1. Predictions on the partial knowledge graph shown in Fig. 7

Performance on short contigs

Because the assembly programs may not generate complete phage genomes from viral metagenomic data, phage host prediction programs should be able to predict hosts on assembled contigs. To evaluate the robustness of CHERRY on short contigs, we generated DNA segments with different length ranges.

First, we generated contigs by cutting the testing viruses' genomes with four different lengths: 5kbp, 10kbp, 15kbp, and 20kbp. We pick a random starting position in the genome and cut a substring of the given length. We repeated this process multiple times until we have sufficient contigs. We finally generated 6,340 short contigs for each length range. We used these short contigs to evaluate VHM-net, vHULK, and CHERRY, which can predict hosts at the species level, and reported the average accuracy in Fig. 8. As the figure shows, although the performance of all methods decreases with the

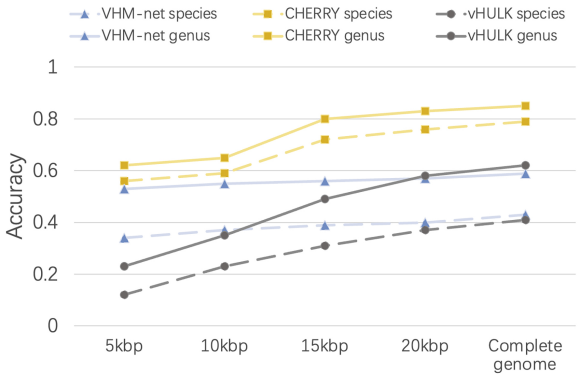

Fig. 8: Prediction performance on short contigs. X-axis: length of the input contigs. Y-axis: accuracy.

decrease of the contigs' length, CHERRY still achieves the best performance under all different length.

Performance on different viral families

Although CHERRY has the best performance in host prediction, its accuracy at species is slightly lower than 0.8. In order to identify the reasons, we conducted a closer examination of the performance for different viral families. Caudovirales, which contains phages with tails, is the order with the most sequenced prokaryotic viruses in the RefSeq database. The test set is dominated by three families under *Caudovirales*: *Siphoviridae*, *Myoviridae*, and *Podoviridae*. Thus, we group the phages by their family taxonomy and record the host prediction results accordingly.

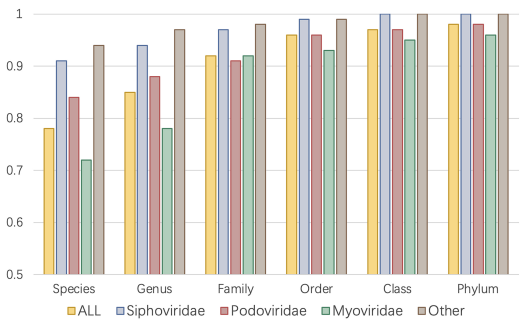

Fig. 9: Host prediction results on different groups of viruses. X-axis: different taxonomic rank. Y-axis: accuarcy. ALL: the accuracy on the whole dataset, which is the same as Fig. 4 (B). Other: accuracy of viruses that do not belong to *Caudovirales*.

As shown in Fig. 9, The accuracy of viruses that do not belong to *Caudovirales* is always the best. The performance of phages in *Mydoviridae* is worse than other groups at species and genus level but increases largely at the family level. One possible reason is that, as discussed in [17, 31], some phages in *Myoviridae* have the potential to infect multiple hosts from different species and even genera. But in our test set, the positive label for one virus only contains one species. Thus, for the viruses in *Myoviridae*, some of the false predictions might indicate that the virus of interest infect multiple hosts.

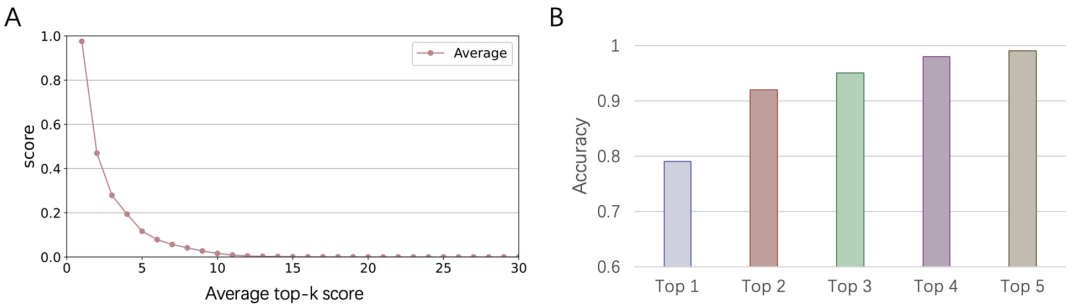

Fig. 10: The experimental results of top-k prediction. A: Tendency of the prediction score. X-axis: the sorted index by k. Y-axis: average score of the top-k prediction. B: The accuracy using top-k prediction..

Top k prediction scores

Given the possible multiple hosts for some viruses, we also provide an alternative evaluation metric based on top k predictions. Instead of only reporting the prokaryote with the highest prediction score, CHERRY also allows users to output multiple hosts. The experiment shows that the known interactions are usually contained in top k outputs. In this section, we will first show the tendency of the prediction score. Because there are 60,105 candidate prokaryotic genomes for each testing virus, the decoder will score each virus-prokaryote pair, leading to 60,105 sorted scores for each testing virus. We will calculate the average of the highest score for all testing viruses, the second highest score, etc. Because there are 60,105 values, we only show the highest 30 scores in Fig. 10(A).

As Fig. 10(A) shows, the average score drops precipitously after the first ten values, suggesting that CHERRY has a strong selection preference for a few possible hosts. In Fig. 10(B), we present the top-k host prediction accuracy using the 60,105-dimensional score vector. Specifically, if the real host label of a testing virus exists in the highest k predictions (scores), we will consider this as a correct prediction for the top-k accuracy. As shown in Fig. 9(B), the top-2 accuracy increases largely from top-1, and the growth trend becomes slower after that. We also found that even if the scores of some real virus-host interactions are not the top 1 score, their scores are usually larger than 0.9, indicating that they are predicted with high confidence. Thus, CHERRY can support a method for further improving the host prediction results: using score threshold and outputting top k predictions. Although this method might predict more than one virus-prokaryote interaction for a given virus, it can largely narrow the search scope of the potential host.

Host prediction on metagenomic data

In this section, we will validate CHERRY on host prediction for possibly novel viruses from metagenomic data. We choose two newly published metagenomic datasets containing viruses in two habitats: glacier [50] and human gut [8]. The authors used assembly tools and virus identification tools, such as VirSorter [41], to obtain virus-originating contigs from the samples. Then, we applied CHERRY to predict hosts for these virus contigs.

Case study one: newly identified viruses in glacier metagenomic data

This data set was sequenced from the core of the glacier [50]. Due to global warming, the melting glacier might release those ancient viruses to the environment in the future. Metagenomic sequencing provides a powerful means to study the virus composition.

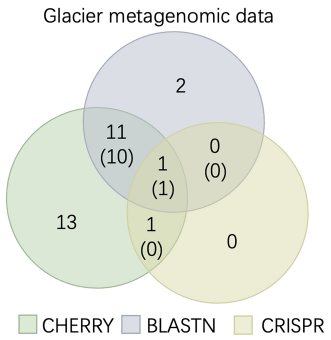

Fig. 11: Host prediction on the glacier metagenomic data. The numbers without parentheses represent the number of viruses. The numbers with parentheses represent the number of viruses with the same predicted hosts. For example, out of 12 viruses with predictions by CHERRY, 12 also have predictions with BLASTN. And 11 of them have the same host prediction.

The authors reported 33 highly confident virus contigs from the metagenomic data. The length of the contigs range from 12,041 to 93,811. According to the authors' analysis, these metagenomic data contains four dominant bacterial genera including *Methylobacterium*, *Sphingomonas*, *Janthinobacterium*, and *Herminiimonas* and 3 putative laboratory contaminants including *Synechococcus phages*, *Cellulophaga phages*, and *Pseudoalteromonas phages*. Thus, we use the bacteria under these genera with all prokaryotes in our database as candidate hosts and run CHERRY.

Because the authors already reported the host prediction using BLASTN and CRISPR, we re-used their predictions and compared that with our method. To report a more precise prediction, we only keep the predictions of prokaryotes with

a score larger than 0.9. Thus, some viruses do not have predicted hosts. The Venn diagram of the three methods is shown in Fig. 11. The overlap region means the number of viruses predicted by both or all methods. The number in the parentheses represents the number of viruses with the same predicted hosts. As shown in Fig. 11, CHERRY can predict hosts for more viruses (near 80%) on this dataset. This is expected because our previous analysis and experiments have shown that BLASTN and CRISPR can only predict hosts for a very limited number of viruses. For the viruses with predictions from either BLASTN or CRISPR, the prediction of CHERRY is largely consistent with them. Specifically, among the 14 BLASTN-predicted viruses and 2 CRISPR-predicted viruses, CHERRY has 11 identical predictions as BLASTN and one identical prediction as CRISPR.

Case study two: newly identified viruses in gut metagenomic data

In a newly published human gut metagenomic study [8], the authors identified 3,738 complete phage genomes that represent 451 putative genera. Investigating the host of these viruses will extend our understanding of how these newly identified viruses affect human health.

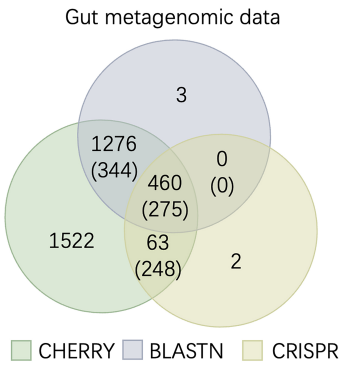

Fig. 12: Host prediction on the gut metagenomic data. The numbers without parentheses represent the number of viruses. The numbers with parentheses represent the number of viruses with the same predicted hosts.

We reused the reported results of CRISPR from [8] and ran BLASTN and CHERRY. Since there is no prior information of the prokaryotes in the this metagenomic data, we use all prokaryotes in our database to construct a graph. The result is shown in Fig. 12. CHERRY significantly improves the number of predicted viruses compared to BLASTN and CRISPR. Up to 89% (3,321/3,738) viruses were predicted by CHERRY with a score threshold of 0.9. What's more, the predictions of CHERRY are highly consistent with the CRISPR results. Only two virus contigs predicted by CRISPR have no predictions by CHERRY because their scores are less than the threshold. All other CRISPR-predicted contigs (460+63) have the same labels (275+248) as CHERRY. We also found that although BLASTN output hosts for 2,131 (57%) viruses, many have multiple alignments. Only 59% (275/460) of the BLAST predictions are consistent with CRISPR. Considering that CRISPR has high precision but low recall, CHERRY's output is consistent with our previous experiments, demonstrating both high recall and precision.

Predicting viruses that infect prokaryotes

The framework of CHERRY is to estimate how likely there is a link (infection) between a virus p_i and a prokaryote h_i. Thus, unlike available tools, CHERRY can also be used to output viruses that infect a prokaryote of interest. We use Eq. 9 to predict the viruses that infect given prokaryotic genomes. CHERRY will take prokaryotic genomes as input and output viruses with prediction scores above a given threshold. This function will be helpful when users want to find candidate phages that can infect and kill antibiotic-resistant bacteria. We will use *recall* and *precision* introduced in the *Experimental setup* Section to evaluate the performance.

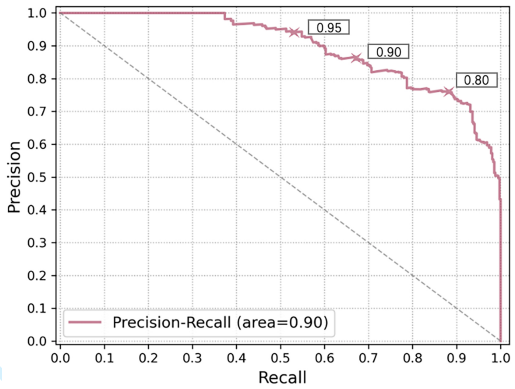

Fig. 13: The precision-recall curve of predicting viruses infecting targeted prokaryotes. X-axis: recall, Y-axis: precision. The performance for three thresholds 0.95, 0.9, and 0.8 are marked with the cross sign on the curve.

As shown in Fig. 13, we draw the precision-recall curve by recording the precision and recall under different thresholds. When using a more lenient threshold, the recall increases with a sacrifice of the precision decreases. Users can choose the thresholds according to their needs. In order to achieve high precision, we use 0.9 as the default threshold in our program.

Case study: antibiotic-resistant bacteria

Phage therapy is one strategy to treat infections of superbugs, which developed resistance to antibiotics. Because many phages have a narrow range of infection, cocktail therapy is commonly used to increase the efficacy. In this case study, we demonstrate how CHERRY can be used to provide candidates for the cocktail therapy.

We test CHERRY on three different antibiotic-resistant bacterial strains: *Staphylococcus aureus subsp. aureus RN4220*, *Klebsiella pneumoniae KPNIH1*, and *Mycobacterium tuberculosis HN-506*. Because our goal is to identify viruses infecting these strains, there is no need to use all the prokaryotes in RefSeq. Instead, we only need to create a graph containing all viruses in the RefSeq, their hosts, and the three bacterial strains. Thus, our knowledge graph contains 1,943 viruses and

226 prokaryotes. The training set is the same as the previous experiments, so only viruses and prokaryotes released before 2015 have labels when training. After training, we tested whether CHERRY can predict viruses that infect these three prokaryotes. Because we have 1943 viruses, we can generate a total of $1943 * 3$ virus-prokaryote pairs for validation. As described in the previous section, we use 0.9 as the cutoff for the final prediction.

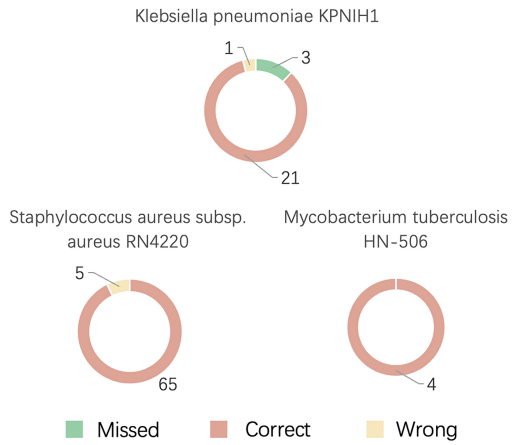

Fig. 14: Performance of predicting viruses infecting *Staphylococcus aureus*, *Klebsiella pneumoniae*, and *Mycobacterium tuberculosis*. 'Missed': CHERRY missed an annotated interaction between a virus and one of the three species. 'Wrong': false positive prediction by CHERRY. 'Correct' represents the number of viruses that infect the prokaryote and their scores are larger than the threshold (predicted correctly).

Because CHERRY was trained at the species level, we will first show the species level performance. Specifically, there are three cases. *Missed* represents that CHERRY miss a true interaction between a virus and one of the three bacteria, which occurs because the prediction score is lower than the given threshold. *Wrong* represents a false positive prediction by CHERRY, indicating that a false interaction receives a score above the threshold. *Correct* represents the true positive prediction by CHERRY.

As shown in Fig. 14, most of the viruses identified by CHERRY indeed infect *Staphylococcus aureus*, *Klebsiella pneumoniae*, and *Mycobacterium tuberculosis*. Also, the precision of *Staphylococcus aureus* and *Mycobacterium tuberculosis* are both 100%. We further investigate which viruses achieve the highest score for each strain. The results show that both *Staphylococcal phage MR003* and *Klebsiella phage PhiKpNIH-2* has the highest score among 1943 virus-prokaryote pairs connecting to *Staphylococcus aureus subsp. aureus RN4220* and *Klebsiella pneumoniae KPNIH1*, respectively. The pair between *Mycobacterium phage DS6A* and *Mycobacterium tuberculosis HN-506* has the second highest score. There are some recent researches showing that these three phages infect the three bacterial strains [35, 5, 38]. Our predictions are highly consistent with these experimental evidence, further demonstrating the utility of CHERRY.

Another interesting finding in the experiment is that, as reported in [5], *Mycobacterium smegmatis* is usually used as a carrier bacterium for phage therapy to kill *Mycobacterium tuberculosis HN-506*. As *Mycobacterium smegmatis* is not a human pathogen, it can help the phage move more quickly to find and kill the target of interest, *Mycobacterium tuberculosis HN-506*. In our experiments, we found that the virus-prokaryote pair between *Mycobacterium phage DS6A* and *Mycobacterium smegmatis* can also achieve a high score (0.988), which is higher than the pair between *Mycobacterium phage DS6A* and *Mycobacterium tuberculosis HN-506* (0.978). This prediction is also consistent with the experimental findings.

Discussion

In this work, we propose a new framework, CHERRY, by formulating the host prediction problem as a link prediction problem in a multimodel graph. This multimodal graph consists of different types of prior knowledge, including protein organization, CRISPR, sequence similarity, and k-mer frequency to connect viruses and prokaryote from labelled (training) and unlabelled(testing) data. Then we apply a graph convolutional encoder to embed feature vectors on the graph and use a 2-layer neural network decoder to calculate the probability of whether the query (virus-prokaryote) pair form an infection. We apply the end-to-end training process and apply the negative sampling method to calculate and backpropagate loss. This design helps the model learn features from the training set and test set to enhance the learning ability. The large-scale experiments on 1,940 viruses and 60,105 prokaryotic genomes show that we improve the host prediction accuracy from 43% to 80% at the species level. We also use two case studies to validate the reliability and practicality of our model in real-world applications.

Although CHERRY has greatly improved the performance of host prediction significantly, we have several goals to further optimize or extend CHERRY in our future work. First, CHERRY currently only uses sequence-based features such as sequence similarity and k-mer frequency. Considering the physical interactions between binding proteins, one possible extension is to include protein-protein interactions (PPI) between viruses and prokaryotes. However, because only a few PPIs about prokaryotic viruses are reported, more experiments or computational predictions are needed to augment the graph. Second, CHERRY is trained for species-level host prediction and is not optimized for strain-level host prediction. The high similarity between strains can lead to ambiguous predictions. In addition, another challenge is the fewer training samples at the strain level. We will explore whether CHERRY can be extended for strain-level host prediction in our future work.

Data Availability

All data and codes used for this study are available online or upon request to the authors. The source code of CHERRY is available via: `https://github.com/KennthShang/CHERRY`. The accessions of training set and test set are available via: `https://github.com/KennthShang/CHERRY/Interactiondata`. The

training set is listed in VHM_PAIR_TAX.xls. The test set is listed in TEST_PAIR_TAX.xls.

Conflict of Interest

There is no competing interest.

Funding

City University of Hong Kong (Project 9678241), HKIDS (9360163), and the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA).

References

1. Rodrigo Achigar, Alfonso H Magadán, Denise M Tremblay, María Julia Pianzzola, and Sylvain Moineau. Phage-host interactions in Streptococcus thermophilus: Genome analysis of phages isolated in Uruguay and ectopic spacer acquisition in CRISPR array. *Scientific reports*, 7(1):1–9, 2017.
2. Nathan A Ahlgren, Jie Ren, Yang Young Lu, Jed A Fuhrman, and Fengzhu Sun. Alignment-free oligonucleotide frequency dissimilarity measure improves prediction of hosts from metagenomically-derived viral sequences. *Nucleic acids research*, 45(1):39–53, 2017.
3. Mohammad Al Hasan, Vineet Chaoji, Saeed Salem, and Mohammed Zaki. Link prediction using supervised learning. In *SDM06: workshop on link analysis, counter-terrorism and security*, volume 30, pages 798–805, 2006.
4. Deyvid Amgarten, Bruno Koshin Vázquez Iha, Carlos Morais Piroupo, Aline Maria da Silva, and João Carlos Setubal. vHULK, a new tool for bacteriophage host prediction based on annotated genomic features and deep neural networks. *bioRxiv*, 2020.
5. Taher Azimi, Mehrdad Mosadegh, Mohammad Javad Nasiri, Sahar Sabour, Samira Karimaei, and Ahmad Nasser. Phage therapy as a renewed therapeutic approach to mycobacterial infections: a comprehensive review. *Infection and drug resistance*, 12:2943, 2019.
6. Daniel N Baker and Ben Langmead. Dashing: fast and accurate genomic distances with HyperLogLog. *Genome biology*, 20(1):1–12, 2019.
7. Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: an open source software for exploring and manipulating networks. In *Third international AAAI conference on weblogs and social media*, 2009.
8. Sean Benler, Natalya Yutin, Dmitry Antipov, Mikhail Rayko, Sergey Shmakov, Ayal B Gussow, Pavel Pevzner, and Eugene V Koonin. Thousands of previously unknown phages discovered in whole-community human gut metagenomes. *Microbiome*, 9(1):1–17, 2021.
9. Charles Bland, Teresa L Ramsey, Fareedah Sabree, Micheal Lowe, Kyndall Brown, Nikos C Kyrpides, and Philip Hugenholtz. CRISPR recognition tool (CRT): a tool for automatic detection of clustered regularly interspaced palindromic repeats. *BMC bioinformatics*, 8(1):1–8, 2007.
10. Dimitri Boeckaerts, Michiel Stock, Bjorn Criel, Hans Gerstmans, Bernard De Baets, and Yves Briers. Predicting bacteriophage hosts based on sequences of annotated receptor-binding proteins. *Scientific reports*, 11(1):1–14, 2021.
11. Benjamin Bolduc, Ho Bin Jang, Guilhem Doulcier, Zhi-Qiang You, Simon Roux, and Matthew B Sullivan. vConTACT: an iVirus tool to classify double-stranded DNA viruses that infect Archaea and Bacteria. *PeerJ*, 5:e3243, 2017.
12. Benjamin Buchfink, Chao Xie, and Daniel H Huson. Fast and sensitive protein alignment using DIAMOND. *Nature methods*, 12(1):59–60, 2015.
13. D Burstein, CL Sun, CT Brown, I Sharon, K Anantharaman, AJ Probst, BC Thomas, and JF Banfield. Major bacterial lineages are essentially devoid of CRISPR-Cas viral defence systems. Nat Commun 7: 10613, 2016.
14. Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. BLAST+: architecture and applications. *BMC bioinformatics*, 10(1):1–9, 2009.
15. Alessandra Carbone. Codon bias is a major factor explaining phage evolution in translationally biased hosts. *Journal of Molecular Evolution*, 66(3):210–223, 2008.
16. Eoghan Casey, Douwe Van Sinderen, and Jennifer Mahony. In vitro characteristics of phages to guide 'real life'phage therapy suitability. *Viruses*, 10(4):163, 2018.
17. Sandra Chibani-Chennoufi, Anne Bruttin, Marie-Lise Dillmann, and Harald Brüssow. Phage-host interaction: an ecological perspective. *Journal of bacteriology*, 186(12):3677–3686, 2004.
18. Clément Coclet and Simon Roux. Global overview and major challenges of host prediction methods for uncultivated phages. *Current Opinion in Virology*, 49:117–126, 2021.
19. Felipe Hernandes Coutinho, Asier Zaragoza-Solas, Mario López-Pérez, Jakub Barylski, Andrzej Zielezinski, Bas E Dutilh, Robert Edwards, and Francisco Rodriguez-Valera. RaFAH: Host prediction for viruses of Bacteria and Archaea based on protein content. *Patterns*, page 100274, 2021.
20. Patrick Cramer. AlphaFold2 and the future of structural biology. *Nature Structural & Molecular Biology*, pages 1–2, 2021.
21. Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29:3844–3852, 2016.
22. BE Dutilh, N Cassman, K McNair, SE Sanchez, GG Silva, L Boling, JJ Barr, DR Speth, V Seguritan, RK Aziz, et al. A highly abundant bacteriophage discovered in the unknown sequences of human faecal metagenomes. Nat Commun 5: 4498, 2014.
23. Mária Džunková, Soo Jen Low, Joshua N Daly, Li Deng, Christian Rinke, and Philip Hugenholtz. Defining the human gut host–phage network through single-cell viral tagging. *Nature microbiology*, 4(12):2192–2203, 2019.
24. Robert A Edwards, Katelyn McNair, Karoline Faust, Jeroen Raes, and Bas E Dutilh. Computational approaches to predict bacteriophage–host relationships. *FEMS microbiology reviews*, 40(2):258–272, 2016.
25. Robert A Edwards and Forest Rohwer. Viral metagenomics. *Nature Reviews Microbiology*, 3(6):504–510, 2005.
26. Clovis Galiez, Matthias Siebert, François Enault, Jonathan Vincent, and Johannes Söding. WIsH: who is the

host? Predicting prokaryotic hosts from metagenomic phage contigs. *Bioinformatics*, 33(19):3113–3114, 2017.

27. Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.
28. Manolo Gouy and Christian Gautier. Codon usage in bacteria: correlation with gene expressivity. *Nucleic acids research*, 10(22):7055–7074, 1982.
29. Ana Laura Grazziotin, Eugene V Koonin, and David M Kristensen. Prokaryotic Virus Orthologous Groups (pVOGs): a resource for comparative genomics and protein family annotation. *Nucleic acids research*, page gkw975, 2016.
30. Ibtissem Grissa, Gilles Vergnaud, and Christine Pourcel. The CRISPRdb database and tools to display CRISPRs and to generate dictionaries of spacers and repeats. *BMC bioinformatics*, 8(1):1–10, 2007.
31. Sana Hamdi, Geneviève M Rousseau, Simon J Labrie, Denise M Tremblay, Rim Saïed Kourda, Karim Ben Slama, and Sylvain Moineau. Characterization of two polyvalent phages infecting Enterobacteriaceae. *Scientific reports*, 7(1):1–12, 2017.
32. Doug Hyatt, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer, and Loren J Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC bioinformatics*, 11(1):1–11, 2010.
33. Mark Johnson, Irena Zaretskaya, Yan Raytselis, Yuri Merezhuk, Scott McGinnis, and Thomas L Madden. NCBI BLAST: a better web interface. *Nucleic acids research*, 36(suppl_2):W5–W9, 2008.
34. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
35. Sang-Eun Lee, Deog-Yong Lee, Wook-Gyo Lee, B Kang, Yoon Suk Jang, Boyeong Ryu, S Lee, Hyunjung Bahk, and Eungyu Lee. Osong Public Health and Research Perspectives, 2020.
36. Congyu Lu, Zheng Zhang, Zena Cai, Zhaozhong Zhu, Ye Qiu, Aiping Wu, Taijiao Jiang, Heping Zheng, and Yousong Peng. Prokaryotic virus host predictor: a Gaussian model for host prediction of prokaryotic viruses in metagenomics. *BMC biology*, 19(1):1–11, 2021.
37. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
38. Shiri Navon-Venezia, Kira Kondratyeva, and Alessandra Carattoli. Klebsiella pneumoniae: a major worldwide source and shuttle for antibiotic resistance. *FEMS microbiology reviews*, 41(3):252–275, 2017.
39. Joan Carles Pons, David Paez-Espino, Gabriel Riera, Natalia Ivanova, Nikos C Kyrpides, and Mercè Llabrés. VPF-Class: taxonomic assignment and host prediction of uncultivated viruses based on viral protein families. *Bioinformatics*, 2021.
40. Simon Roux, Jennifer R Brum, Bas E Dutilh, Shinichi Sunagawa, Melissa B Duhaime, Alexander Loy, Bonnie T Poulos, Natalie Solonenko, Elena Lara, Julie Poulain, et al. Ecogenomics and potential biogeochemical impacts of globally abundant ocean viruses. *Nature*, 537(7622):689–693, 2016.
41. Simon Roux, Francois Enault, Bonnie L Hurwitz, and Matthew B Sullivan. VirSorter: mining viral signal from microbial genomic data. *PeerJ*, 3:e985, 2015.
42. Jiayu Shang, Jingzhe Jiang, and Yanni Sun. Bacteriophage classification for assembled contigs using graph convolutional network. *Bioinformatics*, 37(Supplement_1):i25–i33, 07 2021.
43. Jiayu Shang and Yanni Sun. Detecting the hosts of bacteriophages using GCN-based semi-supervised learning. *BMC biology*, 19(250):1–15, 2021.
44. Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.
45. Jie Tan, Zhencheng Fang, Shufang Wu, Qian Guo, Xiaoqing Jiang, and Huaiqiu Zhu. HoPhage: an ab initio tool for identifying hosts of phage fragments from metaviromes. *Bioinformatics*, 2021.
46. Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.
47. Weili Wang, Jie Ren, Kujin Tang, Emily Dart, Julio Cesar Ignacio-Espinoza, Jed A Fuhrman, Jonathan Braun, Fengzhu Sun, and Nathan A Ahlgren. A network-based integrated framework for predicting virus–prokaryote interactions. *NAR genomics and bioinformatics*, 2(2):lqaa044, 2020.
48. Weili Wang, Jie Ren, Kujin Tang, Emily Dart, Julio Cesar Ignacio-Espinoza, Jed A Fuhrman, Jonathan Braun, Fengzhu Sun, and Nathan A Ahlgren. A network-based integrated framework for predicting virus–prokaryote interactions. *NAR Genomics and Bioinformatics*, 2(2), 06 2020.
49. Edward Wawrzynczak. A global marine viral metagenome. *Nature Reviews Microbiology*, 5(1):6–6, 2007.
50. Zhi-Ping Zhong, Funing Tian, Simon Roux, M Consuelo Gazitúa, Natalie E Solonenko, Yueh-Fen Li, Mary E Davis, James L Van Etten, Ellen Mosley-Thompson, Virginia I Rich, et al. Glacier ice archives nearly 15,000-year-old microbes and phages. *Microbiome*, 9(1):1–23, 2021.

Jiayu Shang received his bachelor's degree from Sun Yat-sen University. He is now pursuing his Ph.D. degree at the City University of Hong Kong. His research interest is bioinformatics, with a focus on algorithm design for analyzing microbial sequencing data.

Yanni Sun is currently an associate professor in the Department of Electrical Engineering at the City University of Hong Kong. She got her Ph.D. in Computer Science and Engineering from Washington University in Saint Louis, USA. Her research interests are sequence analysis and metagenomics.