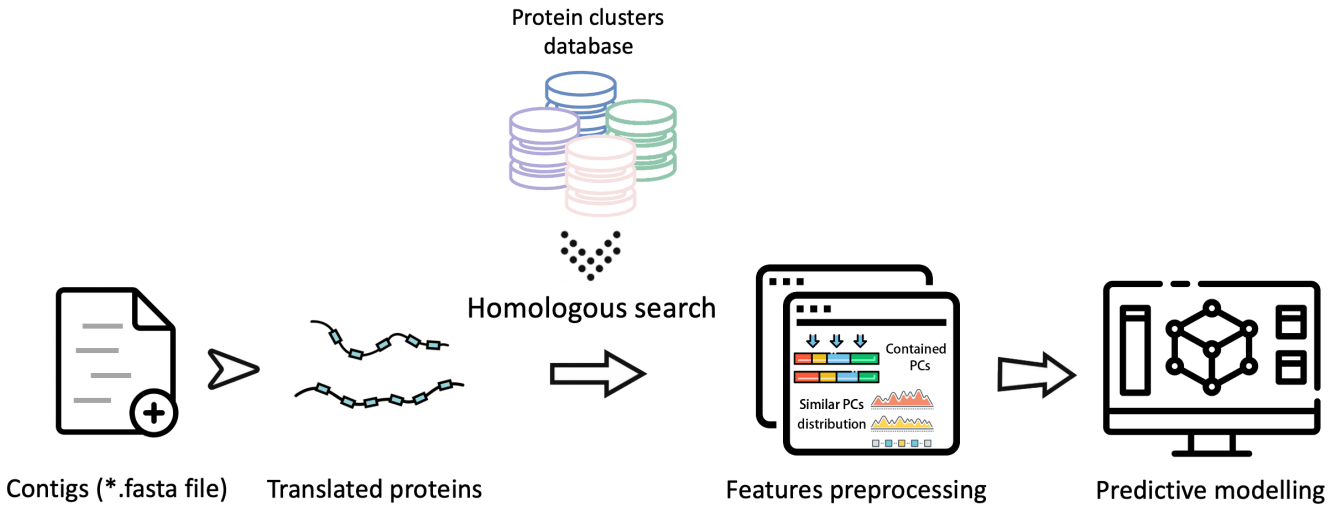
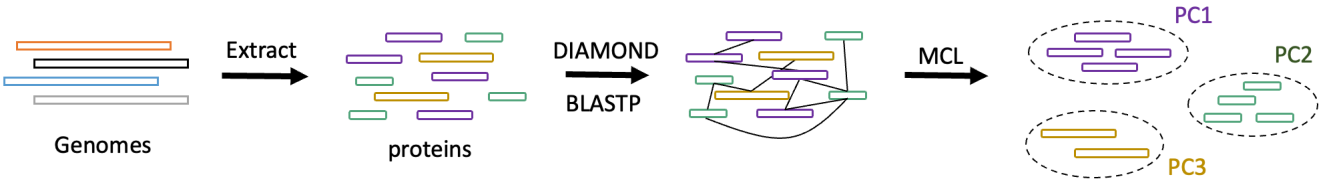
Components of the PhaSUITE

The overall workflow of PhaSUITE is presented in terms of data collection and curation, feature encoding, and model prediction as shown in the figure below:



Construction of protein cluster database



The protein clusters (PCs) are constructed on the phage sequence and the precedure can be listed as below:

- In order to be consistent with the gene prediction process of the query sequences, we apply gene finding and protein translation for the DNA genomes using Prodigal.
- We run all-against-all DIAMOND BLASTP on the predicted proteins. Protein pairs with alignment E-value less than 1e-3 are used to create a protein similarity network, where the nodes represent proteins and the edges represent the recorded alignments. The edge weight encodes the corresponding alignment's E-value.
- Markov clustering algorithm (MCL) is employed to group similar proteins into the clusters using default parameters. All the clusters that contain fewer than two proteins are removed.

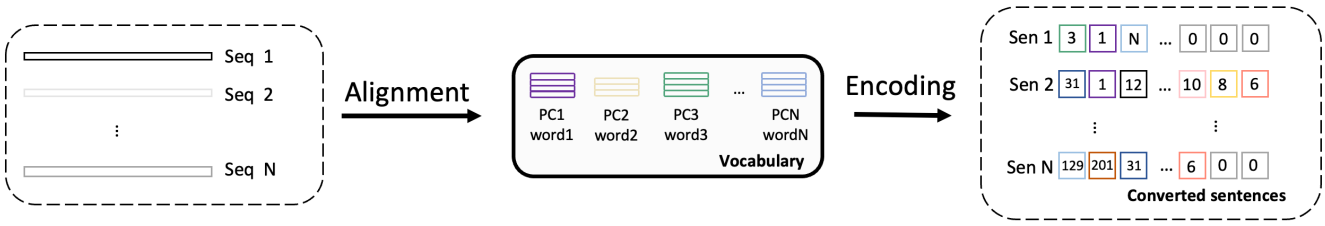
Feature encoding

The feature encoding of our tools is based on the alignment results against protein cluster database. There are two major features derived from the PCs alignment:

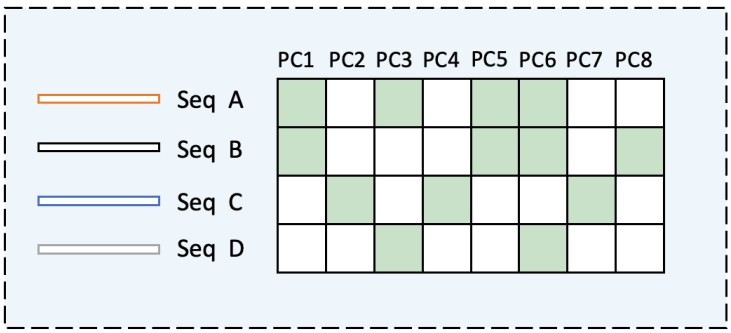
1. Protein cluster sentences
2. Protein cluster sharing network

For each query contig, PhaSUITE will run **Prodigal** for protein translation. Then, DIAMOND BLASTP is employed to search against the protein cluster database. Then:

Protein cluster sentences: PhaSUITE will identify the matched protein clusters for the translated proteins by conducting similarity search. To be specific, it will identify the reference protein incurring the smallest E-value and assign the query with this reference protein's cluster. PhaSUIT will record both the ID of the PC and the position of the protein in the query contig to construct the PC sentences.



Protein cluster sharing network: PhaSUITE will calculate the expected number of sharing at least an observed number of proteins clusters between query contigs and reference genomes. To be specific, PhaSUITE computes the probability that any two sequences containing *a* and *b* protein clusters share at least *c* clusters. Then, PhaSUITE can connect contigs and reference genomes by measuring whether the similarity between two seqeunces is significance to construct the PCs sharing network.



Example (A and B):

$c = 3, n = 8, a = b = 4$

Share at least *c* protein clusters

$$P(X \geq c) = \sum_{i=c}^{\min(a,b)} \frac{C_a^i C_{n-a}^{b-i}}{C_n^b}.$$

Model construction

Each subprogram utilize different

PhaMer

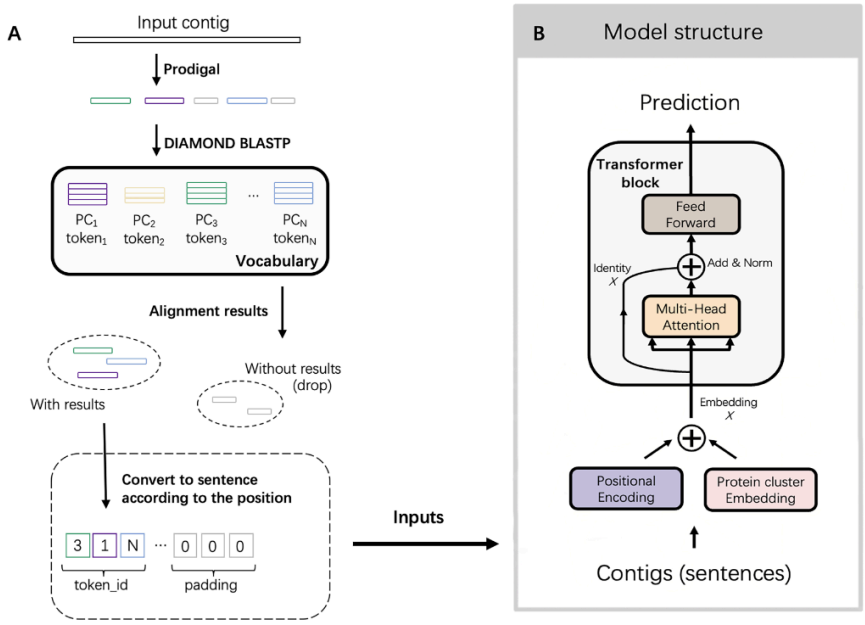


Fig. 1: The pipeline of PhaMer. (A): converting inputs into protein-token sentences. During training we apply Prodigal to predict open reading frames (ORFs) from training phages and translate ORFs into proteins. Then a clustering method is applied to generate protein clusters (PCs), which are the tokens in Transformer. During the test/usage, PhaMer takes contigs as input and convert them into protein-token sentences. (B) Transformer network architecture. The converted sentences are fed into the Transformer model and a prediction is made.

PhaMer employ a contextualized embedding model from natural language processing (NLP) to learn protein-associated patterns in phages. Specifically, by converting a sequence into a sentence composed of protein-based tokens, we employ the embedding model to learn both the protein composition and also their associations in phage sequences.

(Fig. 1A) First, we construct the vocabulary containing protein-based tokens, which are essentially protein clusters with high similarities. Then, we apply DIAMOND BLASTP [1] to record the presence of tokens in training phage sequences.

(Fig. 1B) Then, the tokens and their positions will be fed into Transformer for contextual-aware embedding. The embedding layer and the self-attention mechanism in Transformer enable the model to learn the importance of each protein cluster and the protein-protein associations. In addition, by using the phages' host genomes as the negative samples in the training data, the model can learn from the hard cases and thus is more likely to achieve high precision in real data.

PhaMer can directly use the whole sequences for training, avoiding the bias of segmentation. We rigorously tested PhaMer on multiple independent datasets covering different scenarios including the RefSeq dataset, short contigs, simulated metagenomic data, mock metagenomic data, and the public IMG/VR dataset. We compared PhaMer with four competitive learning-based tools and one alignment-based tool (VirSorter) based on a third-party review [2]. The results show that PhaMer competes favorably against the existing tools.

PhaGCN

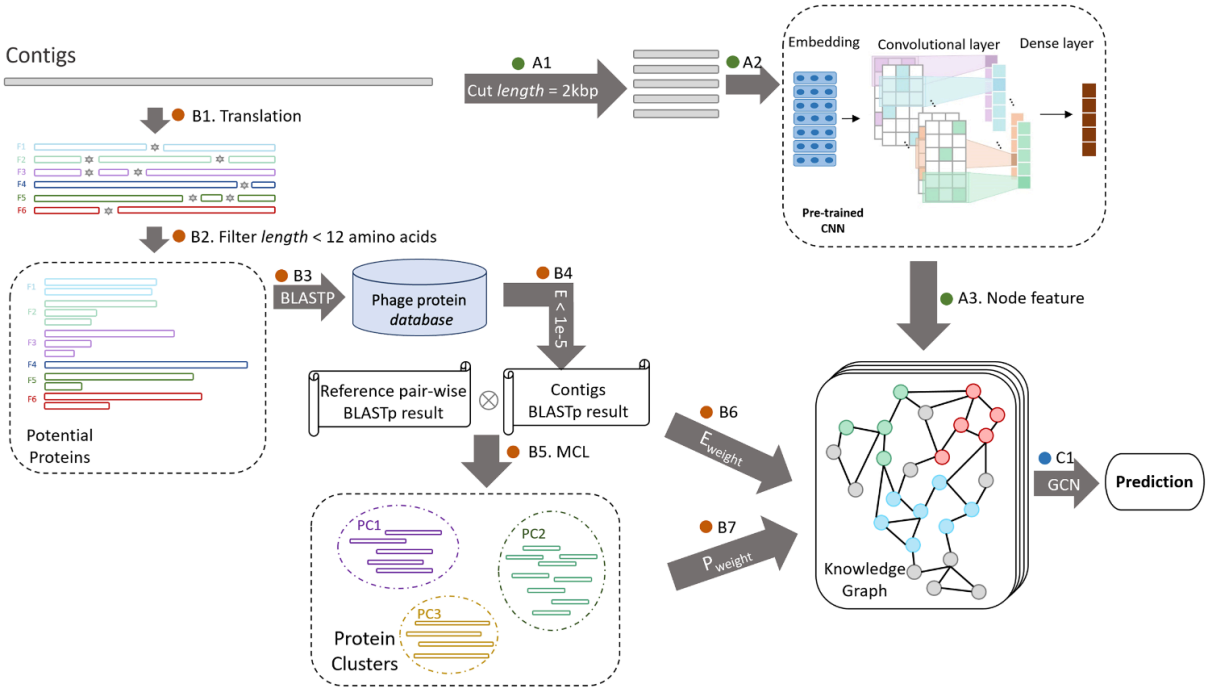


Fig. 1. The pipeline of PhaGCN. A1: cut the contigs into 2kbp segments. A2: feature learning from the inputs using CNN. A3: construct nodes using encoded vectors. B1: contig translation using 6 reading frames. B2: filter short translations (12 amino acids). B3: align contigs against reference database using the DIAMOND BLASTP command. B4: choose the best translated frame for the BLASTP result. B5: use the BLASTP result to construct protein clusters. B6 and B7: define edges based on the sum of the E_{weight} and P_{weight}. C1: construct the knowledge graph for GCN.

Given the enormous diversity of phages and the sheer amount of unlabeled phages, we formulate the phage classification problem as a semi-supervised learning problem. We choose the GCN as the learning model and combine the strength of both the alignment-based and the learning-based methods.

The input to PhaGCN is a knowledge graph. There are two key components in the knowledge graph: node encoding and edge construction. The node is a numerical vector learned from contigs using a CNN. The edge encodes features from both the sequence similarity and the organization of genes.

Fig. 1 contains the major components for node and edge construction.

- **(Fig. 1 A1-A3)** To encode a sequence using a node, a pre-trained convolutional neural network (CNN) is adopted to capture features from the input DNA sequence. The CNN model is trained to convert proximate substrings into vectors of high similarity.
- **(Fig. 1 B1-B4)** The edge construction consists of several steps. We employ a greedy search algorithm to find the best BLASTP results (E-value less than 1e-5) between the translated proteins from the contigs and the database .
- **(Fig. 1 B5)** Then the Markov clustering algorithm (MCL) is applied to generate protein clusters from the BLASTP result .
- **(Fig. 1 B6-B7)** Based on the results of BLASTP (sequence similarity) and MCL (shared proteins), we define the edges between sequences (contigs and reference genomes) using two metrics: P_weight and E_weight.
- **(Fig. 1 C1)** By combining the node's features and edges, we construct the knowledge graph and feed it to the GCN to classify new phage contigs.

We compared PhaGCN with three state-of-the-art models specifically designed for phage classification: Phage Orthologous Groups (POG), vConTACT 2.0, and ClassiPhage. The experimental results demonstrated that PhaGCN outperforms other popular methods in classifying new phage contigs.

PhaTYP

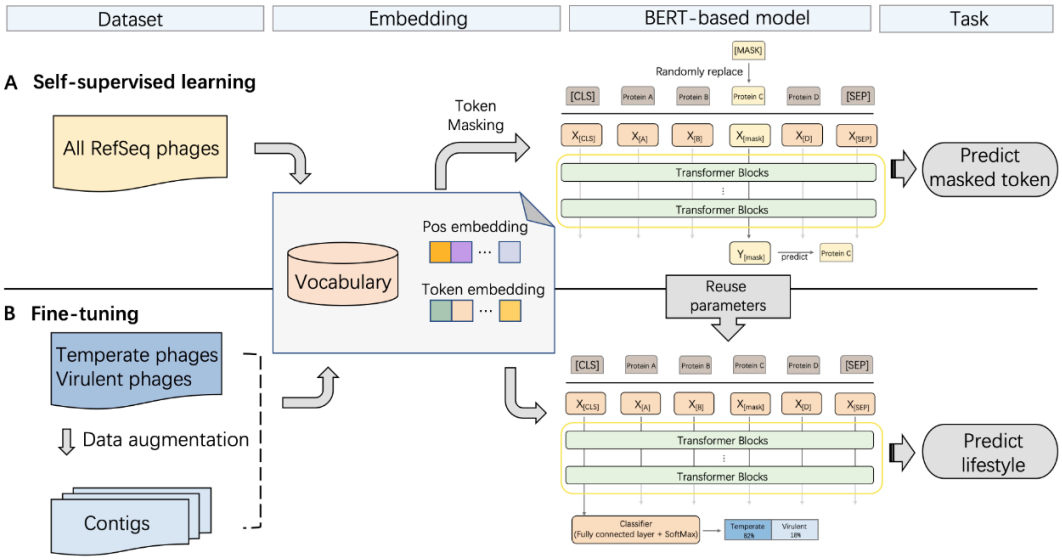


Fig. 1: Two training tasks for PhaTYP using BERT. A: the self-supervised learning task. The input is the masked sentence, and the output is the predicted token at the masked position.‘1 All phage genomes in RefSeq database to train a Mask LM model. B: the fine-tuning task for lifestyle prediction. The pre-trained model is fine-tuned using phages with known lifestyle annotations. The inputs of the model are protein-based sentences, and the outputs are the probabilities of two lifestyle classes: virulent and temperate.

PhaTYP is a BERT-based model that learns the protein composition and associations from phage genomes to classify the lifestyles of phages.

To address the difficulties of classifying incomplete genomes with limited training data, we divide the lifestyle classification into two tasks: a self-supervised learning task (Fig. 1 A) and a fine-tuning task (Fig. 1 B).

- **(Fig. 1A)** To circumvent the problem that only a limited number of phages have lifestyle annotations, we applied self-supervised learning to learn protein association features from all the phage genomes using Masked Language Model (Masked LM), aiming to recover the original protein from the masked protein sentences. This task allows us to utilize all the phage genomes for training regardless of available lifestyle annotations.
- **(Fig. 1B)** In the second task, we will fine-tune the Masked LM on phages with known lifestyle annotations for classification. To ensure that the model can handle short contigs, we apply data augmentation by generating fragments ranging from 100bp to 10,000bp for training.

We evaluated PhaTYP on contigs of different lengths and contigs assembled from real metagenomic data. The benchmark results against the state-of-the-art methods show that PhaTYP not only achieves the highest performance on complete genomes but also improves the accuracy on short contigs by over 10%.

CHERRY

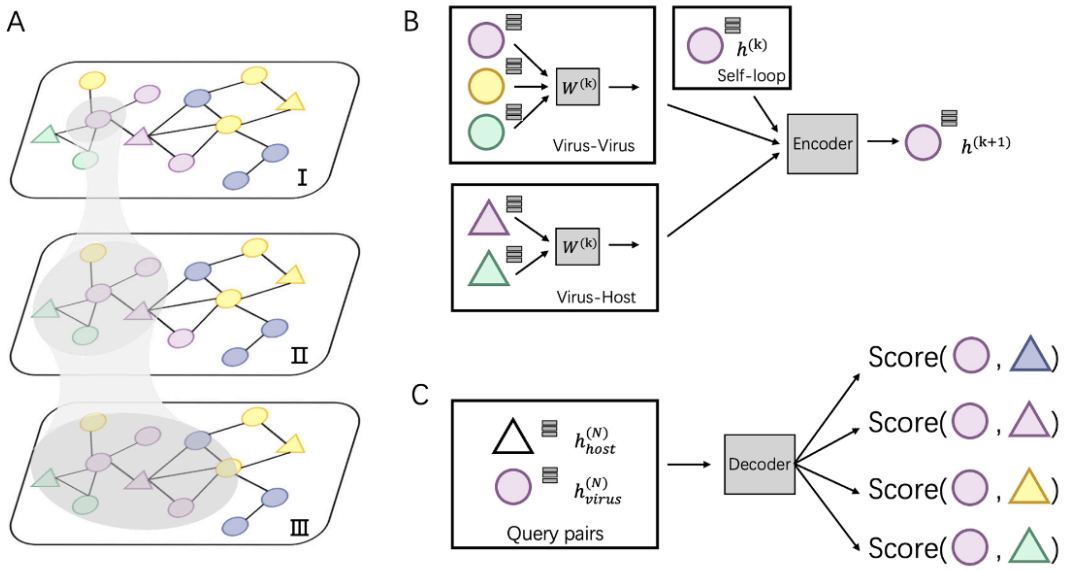


Fig. 1: The key components of CHERRY. A) The multimodal knowledge graph. Triangle represents the prokaryotic node and circle represents virus nodes. Different colors represents different taxonomic labels of the prokaryotes. I-III illustrate graph convolution using neighbors of increasing orders. B) The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

CHERRY can predict the hosts' taxa (phylum to species) for newly identified viruses based on a multimodal graph.

(Fig. 1A) The multimodal graph incorporates multiple types of interactions, including protein organization information between viruses, the sequence similarity between viruses and prokaryotes, and the CRISPR signals . In addition, we use k-mer frequency as the node features to enhance the learning ability.

(Fig. 1B) Rather than directly using these features for prediction, we design an encoder-decoder structure to learn the best embedding for input sequences and predict the interactions between viruses and prokaryotes. The graph convolutional encoder utilizes the topological structure of the multimodal graph and thus, features from both training and testing sequences can be incorporated to embed new node features.

(Fig. 1C) Then, a link prediction decoder is adopted to estimate how likely a given virus-prokaryote pair forms a real infection.

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. Instead of using 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 independent datasets including the RefSeq dataset, simulated short contigs, and metagenomic datasets. We compared CHERRY with WIsH, PHP, HoPhage, VPF-Class, RaFAH, HostG, vHULK, PHIST, DeepHost, PHIAF, and VHM-net. The results show that CHERRY competes favorably against the state-of-the-art tools.

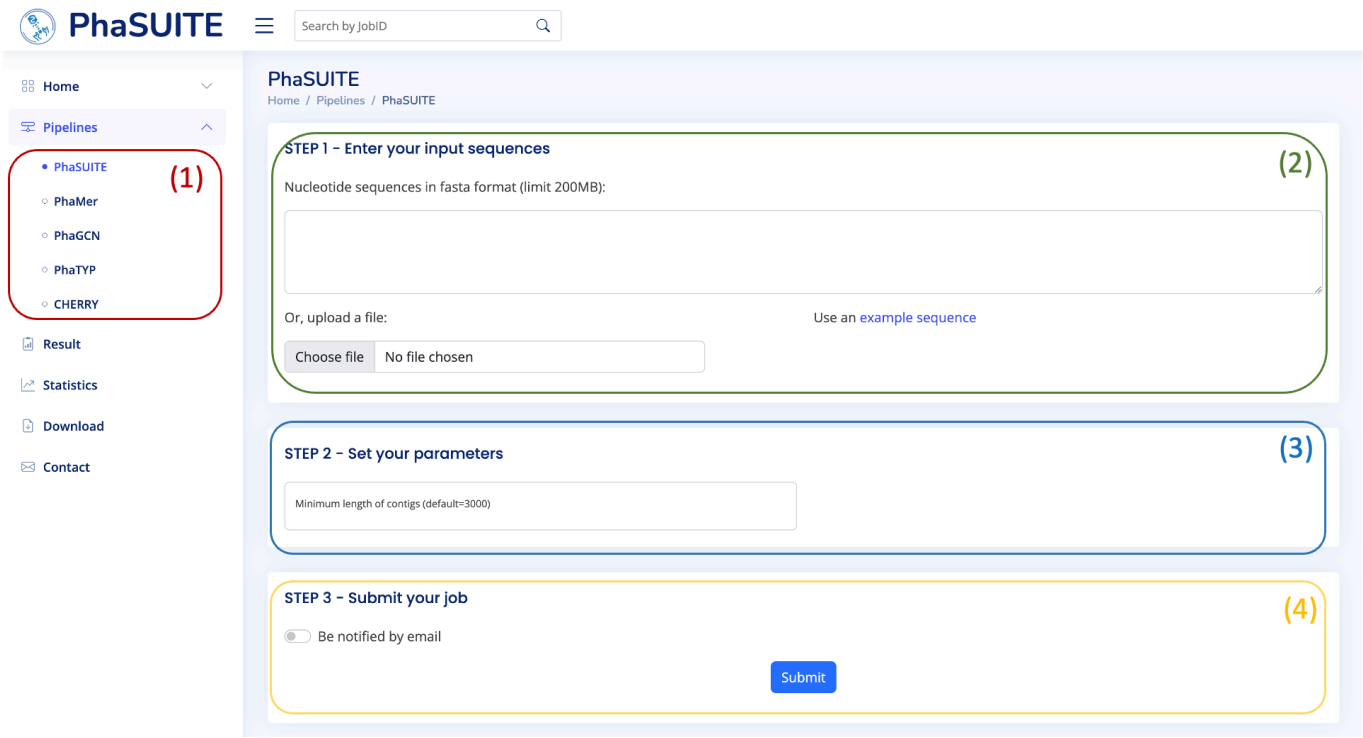
Quick start

PhaSUITE allows users to copy-and-paste or upload their interested DNA contigs in FASTA format in the ***Piplines*** pages. When predicting query contigs, PhaSUIT server provides two options for analysing users' sequences:

1. Running PhaSUITE once for all phage tasks (phage identification, taxa classification, lifestyle prediction, and host prediction).
2. Running subprogram that users are interested in.

However, when running PhaGCN (taxa classification), PhaTYP (lifestyle prediction), and CHERRY (host prediction) in the single-program mode, users should guarantee their inputs sequences are all phages. Otherwise, the results may be influenced by the non-phage sequence. In this case, user can run PhaMer to filter the non-phage genomes first.

Submitting a job



1. Choose a program you want to run in red box:
    - PhaMer: phage identification from metagenomic contigs
    - PhaGCN: family-level taxonomy classification
    - PhaTYP: phages' lifestyle prediction
    - CHERRY: Host prediction
    - PhaSUIT: run the above tools in one pipeline. It will spend less time compared to run tools seperately.
2. Paste or upload your DNA sequences in the green box.
3. Set a threshold for the minimum length of contigs in the blue box. If you want to use the default parameters, then let it blank. The program will only handle the contigs longer than the threshold.

4. Choose whether you want to be notified by email. If yes, turn on the button and paste you email address. Otherwise, submit your task directly and remember the task ID.

Once you submit your job, they webpage will jump to the *Result* page and show the job ID of your submission. The job ID is shown in the red box. Please remember your job ID to find your results.

If you turn on the email notification, you will receive an email **once the job is submitted successfully** and **when the job is finished**. An example is shown as below:

- After submitting the job:

Dear User:

Task PhaSUITE20221120-141046-038385 has been submitted.

Please wait. It will take some time to run the task.

We will notify you by email when the task is complete.

The result will be available in the following link:

Result

Or copy the link and open it in your browser:

https://phage.ee.cityu.edu.hk/result?jobId=PhaSUITE20221120-141046-038385

Thanks for using Phage SUITE.

- When the job is finished:

Dear User:

Task PhaSUITE20221120-142441-089810 is complete.

Click the following link to see the result:

Result

Or copy the link and open it in the browser:

https://phage.ee.cityu.edu.hk/result?jobId=PhaSUITE20221120-142441-089810

Thanks for using Phage SUITE.

The email address used for notification in PhaSUITE is: Phage.SUITE@gmail.com. If you did not receive our email after submitting the job, you may need to check whether it is in your junk mail box. Please add the email into the whitelist if you wish to receive our notification.