READ ME

HoPhage(Host of Phage) is a computational tool that integrates two modules respectively using the deep learning and the Markov chain model to identify the host of a given phage fragment from metagenome or metavirome data at the genus level.

How to use HoPhage

Docker

HoPhage is developed by Python. It was released as a Docker image, so you can easily use HoPhage by installing Docker on your local machine and downloading the pre-built image of HoPhage. Since it is a user-friendly approach to use HoPhage, we highly recommend you use HoPhage by Docker If you are not familiar with configuring the environment of Python.

First, you need to install Docker on your machine (Linux, Windows or MacOS is available) according to the official instructions. After that, you need to download the docker image of HoPhage and create a new docker container. The source code and related files used by HoPhage are all in the `home` folder.

```
docker pull jietan95/hophage:1.1
docker run -it jietan95/hophage:1.1 bash
cd home
```

The only disadvantage of using HoPhage through Docker is that you cannot use GPU. If you do not want to install Docker and you are proficient with Python package installation. You can use HoPhage by installing these acquired dependencies and downloading files of HoPhage by yourself as described below.

Configure the operating environment yourself

Dependencies

The codes of HoPhage are implemented on Python 3.6. To use it, Python 3.6 together with the following packages is required.

- Python 3.6
    - numpy 1.17.4
    - pandas 0.25.3
    - biopython 1.71
    - llvmlite 0.30.0
    - numba 0.46.0
    - scikit-learn 0.23.2
    - pytorch 1.3.0

We recommend using Anaconda to install python and use `conda` or `pip` to install all dependencies except pytorch. Just simply run:

```
pip install numpy==1.17.4
pip install pandas==0.25.3
pip install biopython==1.71
pip install llvmlite==0.30.0
pip install numba==0.46.0
pip install -U scikit-learn==0.23.2
```

The version of pytorch may affect the use of the model, and the version used by HoPhage is 1.3.0. You can just follow the If your machine has a GPU, please configure the corresponding CUDA, CUDNN. Then you can check your CUDA version by `nvidia-smi`, and downloading corresponding pytorch and torchvision, and then install them by `pip`.

```
pip install torch-1.3.0+cu100-cp36-cp36m-linux_x86_64.whl
pip install torchvision-0.4.1+cu100-cp36-cp36m-linux_x86_64.whl
```

Downloading

After you have configured the environment required to run HoPhage, you can download the relevant files as follows:

```
wget http://cqb.pku.edu.cn/ZhuLab/HoPhage/HoPhage_V_1_1.zip
unzip HoPhage_V_1_1.zip
mv HoPhage_V1_1 HoPhage
cd HoPhage
```

The source code and related files used by HoPhage are all in this compressed file.

Data preparation

Before using HoPhage to predict the host of phage fragments, you need to do some data preparation,

- PPR-Meta

  PPR-Meta is designed to identify metagenomic sequences as phages, chromosomes or plasmids. If you are getting metagenomic data rather than metavirome data, it is recommended to use PPR-Meta or other tools that can identify short phage sequence fragments to screen out the phage sequence.

- Prodigal

  Prodigal is a fast and reliable protein-coding gene prediction tool. Although prodigal is designed for prokaryotic genomes, we have found that it has the best performance on short phage sequence fragments during actual use, so we still recommend using it for gene prediction of phage fragments in metagenome or metavriome.

- taxize

  Taxize is an R package for taxonomic information from around the web. If you want to use HoPhage with your candidate host range, a genera list are required to provide. Since you may lack the taxonomy information of prokaryotes and only have its GenBankID or RefSeqID, taxize is recommended to use for batch processing.

```
library(taxize)
# Get NCBI taxonomy UID from GenBankID.
uid <- genbank2uid(id='AJ748748')        # GenBank accession number, gi number, RefSeq accessior
# Retrieve the taxonomic hierarchy for a given taxon UID.
temp <- classification(uid, db = 'ncbi')
ind <- which(temp[[1]]$rank=='genus')    # genus as an example
genus <- temp[[1]]$name[ind]
```

Usage

```
python predict.py [-h] -q QUERY_PHAGE -c QUERY_PHAGE_CDS [-o OUTPUT_DIR]
              [-w WEIGHT_HOPHAGE_S] [-g GENUS_RANGE] [--all]
```

Options

```
-h, --help          show this help message and exit
-q QUERY_PHAGE      A directory containing all single query phage fragments with .fasta/.fna suffix
                    OR a file with .fasta/.fna suffix which contains all query phage fragments
-c QUERY_PHAGE_CDS  A directory containing all cds output files with .fasta/.fna suffix
                    of single query phage fragment predicted by Prodigal
                    OR a file with .fasta/.fna suffix
                    which contains all cds output of query phage fragments
-o OUTPUT_DIR       Output directory. The default is the current path
-w WEIGHT_HOPHAGE_S     Weight of HoPhage-S. Default = 0.5
-g GENUS_RANGE      A file containing host genera names of interest
--all               If set, scores of all genera will be outputted
```

Example

If you use HoPhage by Docker, you can run a just simple example from the existing file:

```
cd Prodigal-2.6.3
prodigal -i ../examples/phage_frag.fna -d ../examples/phage_cds.fna -p meta
cd ..
mkdir output_example
python predict.py -q examples/phage_frag.fna -c examples/phage_cds.fna -o output_example -w 0.5 -g exa
```

If you configure the operating environment yourself and use HoPhage, you need to download and install Prodigal before you predicting by HoPhage. After you obtain the gene annotation file of all query phage fragments, you can run the last two lines in the code above.

It is worth noting that when you run HoPhage in an environment configured by yourself, you can specify files in any path on your machine as input or output prediction results to any folder. But when you use Docker to run HoPhage, you need to copy the input file to the container.

```
docker cp local_path container_ID:/container_path
```

When the prediction is completed, you also need to manually copy the output result to the outside of the container, that is, any folder on your local machine.

```
docker cp container_ID:/container_path local_path
```

Notice: the Plog_all.npy in this GitHub link doesn't all pre-calculated Markov chain models. If you need our pre-calculated Markov chain models of all prokaryotes in our data set, you can download "HoPhage_S_Plog_all.zip" at "http://cqb.pku.edu.cn/ZhuLab/HoPhage/data/" and unzip it to the folder "model".

In addition, users can use "cmm_HoPhage_S.py" to build codon Markov models of some user-defined prokaryotes.

```
python cmm_HoPhage_S.py -i path_prokaryotes -o output_dir
# the input path should contain the genbank files of the prokaryotic genomes
```

Output

The output of HoPhage consists of 11 columns, representing "ID" (ID of the query phage fragment), "Score-G" (score through HoPhage-G), "Score-S"(score through HoPhage-S), "Integrated Score"(weighted average of the two scores), "Host Name"(GenebankID of the candidate host), the remaining columns are the taxonomic information of this candidate host.

	ID	Score-G	Score-S	Integrated Score	Host Name	Superkindom	Phylum	Class	Order	Family	Genus
2	r1.1	0.870026	-4.07727	0.857826638	LN774769	Bacteria	Firmicutes	Bacilli	Lactobacill	Streptococ	Lactococcus
3	r1.1	0.652292	-4.08367	0.666305147	CP000414	Bacteria	Firmicutes	Bacilli	Lactobacill	Leuconost	Leuconostoc
4	r1.1	0.647765	-4.06863	0.664703486	CP006910	Bacteria	Firmicutes	Bacilli	Lactobacill	Streptococ	Streptococcus
5	r1.1	0.597769	-4.01981	0.628614469	CP014016	Bacteria	Firmicutes	Bacilli	Bacillales	Staphyloc	Staphylococcus
6	r1.1	0.569495	-4.0843	0.593759421	CP013614	Bacteria	Firmicutes	Bacilli	Lactobacill	Enterococ	Enterococcus
7	r1.1	0.486705	-4.04189	0.527969994	CP002453	Bacteria	Bacteroide	Flavobacte	Flavobacte	Flavobacte	Cellulophaga
8	r1.1	0.42725	-4.01129	0.480747496	CP003259	Bacteria	Firmicutes	Clostridia	Clostridiale	Clostridiac	Clostridium
9	r1.1	0.371277	-4.07687	0.421483981	NC_009089	Bacteria	Firmicutes	Clostridia	Clostridiale	Peptostrep	Clostridioides
10	r1.1	0.364884	-4.099	0.412417808	FR687253	Bacteria	Firmicutes	Bacilli	Bacillales	Listeriacea	Listeria

Citation

Jie Tan, Zhencheng Fang, Shufang Wu, Qian Guo, Xiaoqing Jiang, Huaiqiu Zhu. HoPhage: an ab initio tool for identifying hosts of phage fragments from metaviromes.

Contact

If you find any bugs or encounter any problems while using HoPhage, please feel free to contact jie_tan@pku.edu.cn.