Virtifier: A deep learning based identifier for viral sequences from metagenomes

Supplementary information

Yan Miao, Fu Liu, Tao Hou and Yun Liu

September 15, 2021

S1 Methods

S1.1 Seq2Vec

nucleotide sequence:
- ATA*GCC***TGA***AAG*CTTGGATTG
- Seq2Vec dataset:

Table S1. The way of structuring training dataset for codon embedding.

Input	Output		ATA	GCC	TGA	AAG	CTT	GCA	TTG
ATA	GCC	Datapoint1	1	0	0	0	0	0	0
GCC	ATA	Datapoint2	0	1	0	0	0	0	0
TGA	GCC	Datapoint3	0	0	1	0	0	0	0
TGA	AAG	Datapoint4	0	0	1	0	0	0	0
AAG	TGA	Datapoint5	0	0	0	1	0	0	0
AAG	CTT	Datapoint6	0	0	0	1	0	0	0
CTT	AAG	Datapoint7	0	0	0	0	1	0	0
CTT	GGA	Datapoint8	0	0	0	0	1	0	0
GGA	CTT	Datapoint9	0	0	0	0	0	1	0
GGA	TTG	Datapoint10	0	0	0	0	0	1	0

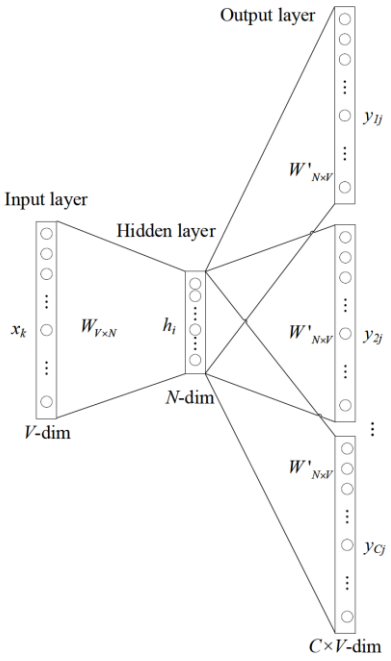


Figure S1. A skip gram model. V is the input dimension. N is the embedding dimension. C is the number of context codons.

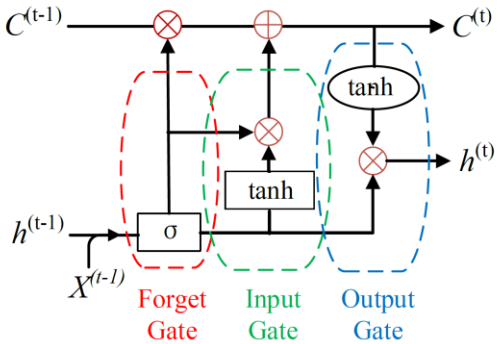


Figure S2. A memory cell in LSTM. Three multiplication units from left to right are input gate, output gate and forget gate respectively.

S1.2 Attention Mechanism

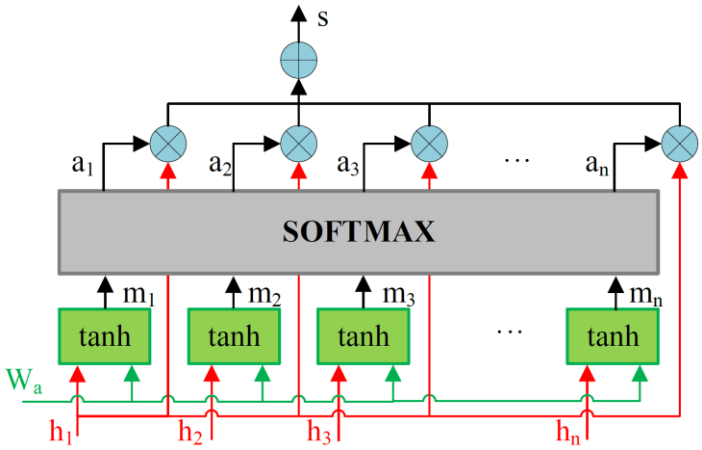


Fig. S3. The attention mechanism over the LSTM network.

The core procedure (Figure S3) of the attention mechanism is to calculate context vector:

$$m_t = \tanh\left(W_a h_t + b\right)$$

where W_a and b_a are weights and bias containing the context information. Then the normalized attention score α_t for each element h_t is calculated by the context vector and another weight matrix W_m through a softmax layer.

$$\alpha_t = \frac{\exp(m_t^T W_m)}{\sum_t \exp(m_t^T W_m)}$$

s is a weighted average of elements in the input sequence:

$$s = \sum_t \alpha_t h_t$$

It is the encoded sequence representation, which can be regarded as an additional contextual feature. Thus, h_t which is closely related to the context generates a large attention score and dominates the final encoding s .

S2 Experiments

Table. S2. The construction of the datasets used in this paper.

Datasets	Number of sequences		Length
	Virus	Host	
Training	4,500	4,500	500bp
Testing	500	500	500bp
CAMI_high	46	1.051	100-500bp
Real human gut samples for short sequences	301	253	100-500bp
	726	602	500-3,000bp
Real human gut samples for long sequences	477	383	3,000-5,000bp
	267	282	>5,000bp

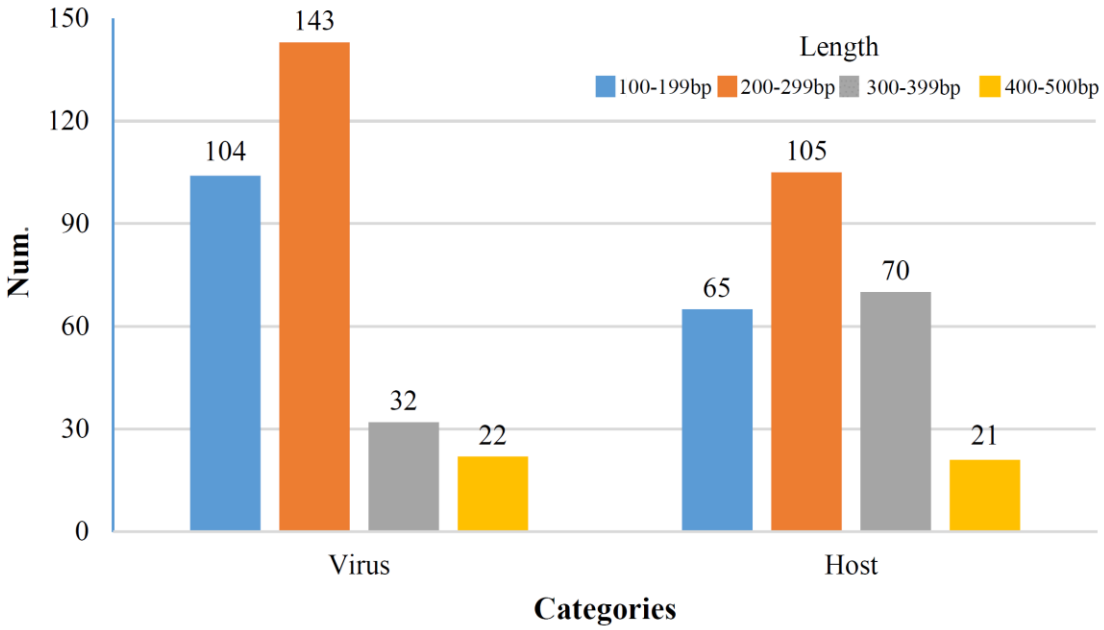


Fig. S4. The number of sequences with various length in the real human gut metagenomics data.

S2.1 Performance on sequences of different length in training and testing dataset

The query test sequences in general got highest AUROC scores when the sequence length in test matched those on which the network was trained. The performance dropped dramatically when the sequence length in the training stage was much longer than query sequence length. Thus, sequences of 300bp and 500bp length were respectively used to train Virtifier. The two trained models were separately utilized to predict sequences <300bp and 300-500bp.

Sequences of such short length were always hard or low-accurately to be identified. LSTM network had a state line bypassing all of the units to remove some of the vanishing gradients problems, which made it able to learn a lot of longer term information than RNN. However, there still existed a sequential path from older past cells to the current one. It was even more complicated because of its additive and forget branches attached. As a result, the limitation led to a bad performance to remember a much longer sequential information. This may be why our LSTM model performed better as the sequence length shortened. When the sequences longer than 1,000bp were used to evaluate our model, the performance reduced dramatically no matter what length of sequences were used for training and testing. To ensure performance, Virtifier was specially utilized to identify viruses with length shorter than 500bp.

S2.2 The effect of 'word' length and embedding size

In the paper, a nucleotide sequence was converted into a string of codons (three bases) with a stride of one base. It was an intuition that the gene expression in a nucleotide sequence was realized by codons, which played a key role in the synthesis of amino

acids. To confirm if the intuition played a positive role in viral identification, every sequence with length of 500bp in our dataset was split into a string of 'words' with two, three, four and five bases. The corresponding lengths of dictionary were 16, 64, 256 and 1024, respectively. Then a series of embedding sizes (8, 20, 30, 100, 150, 300 and 500) were selected to cross-validate with dictionary lengths. When the dictionary length was larger than embedding size, the corresponding model was trained and fine-tuned to obtain an AUC value. The validation result was shown in Figure S5.

In practice, the number of bases in a 'word' was chosen as three (number of bases in a codon), and the embedding size was set as 20 (number of amino acid totally), which may achieve a better performance.

S2.3 Visualization of the embedding performance

To train a representative embedding matrix, virus RefSeq genomes collected from NCBI as supplied in VirFinder were all input to the embedding layer. t-SNE (t-distributed stochastic neighbor embedding), with the ability to maintain local structure, was used to plot a 2D scatterplot so that the spatial location relationships among embedded codon vectors can be visualized (Figure S6) for the reason that points with similar distance in the high-dimensional data space were still similar in the low-dimensional data space.

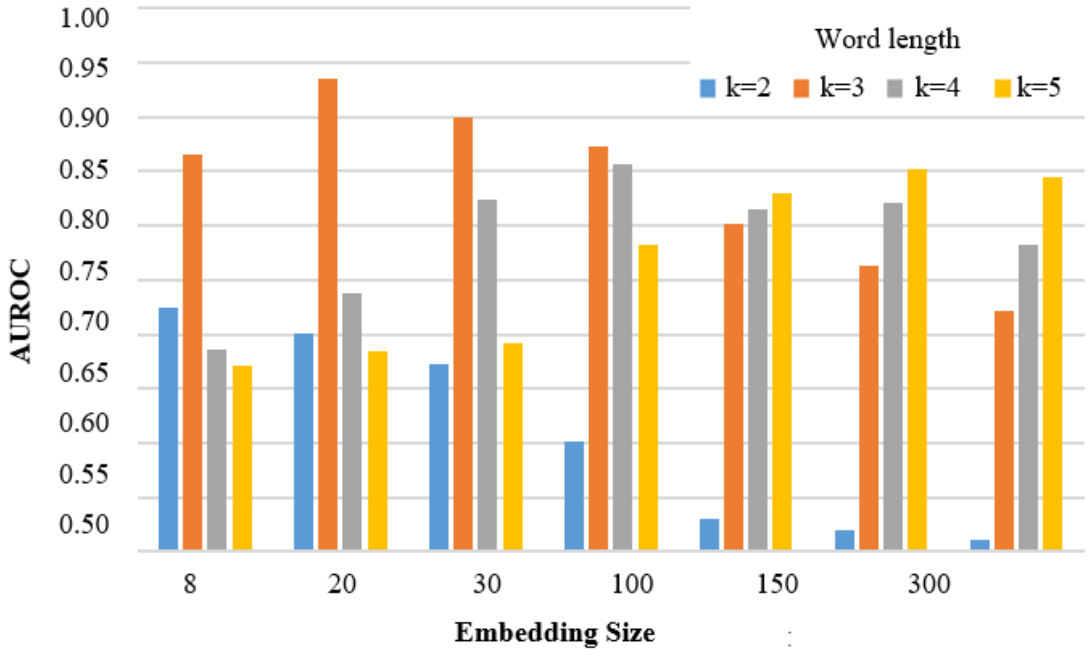


Fig. S5. The effect of 'word' length and embedding size on the performance of embedding.

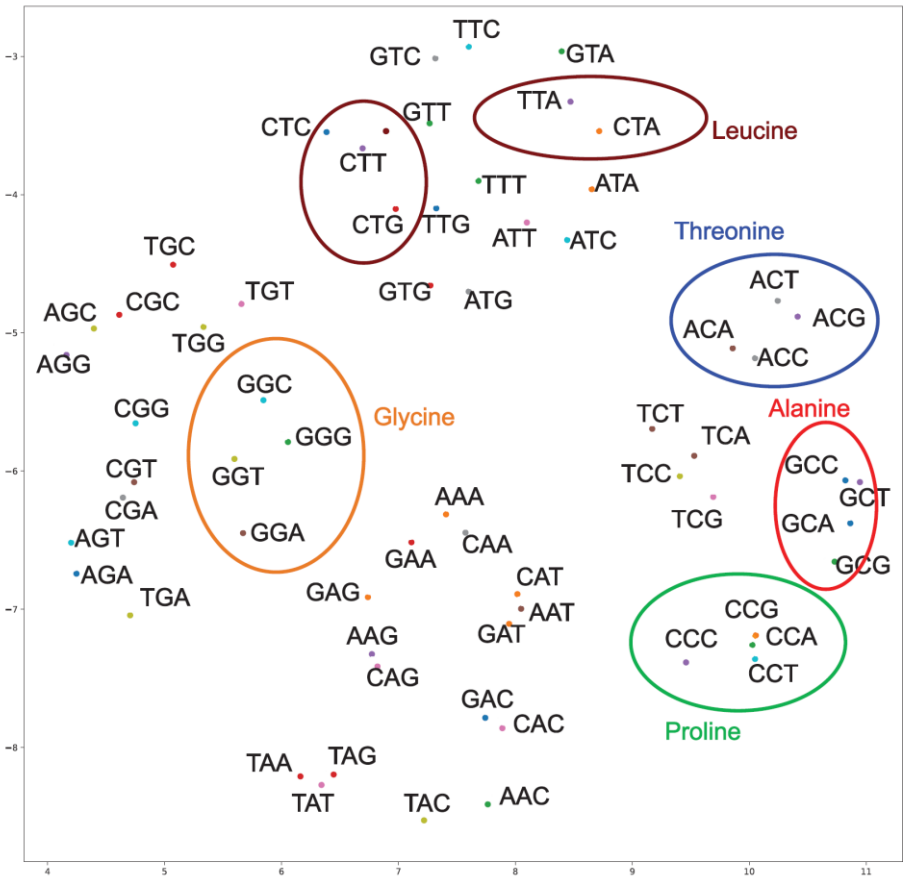


Fig. S6. The visualization of the spatial location relationships among embedded codon vectors.

It can be seen that some codons representing the same amino acid were relatively close in distance, such as GCC, GCT, GCA and GCG representing Alanine, ACT, ACG, ACA and ACC representing Threonine, CCG, CCA, CCT and CCC representing Proline and so on. There were a few codons such as AGT and AGC having a relatively longer distance in the 2D scatterplot from TCT, TCA, TCC and TCG that represent the same amino acid (Serine), which was a special case. It might be caused by the overlapped one-base-stride division from codon sentence to a string of codons. Some divided codons may not really exist during representing amino acids.

S2.4 Results on the three datasets

Table S3. Comparison of Recalls, Precisions and F1 Scores of the four methods on 300bp and 500bp test dataset. Bold values represent the best performance.

Criteria		VirFinder	DeepVirFinder	PPR-Meta	Virtifier
300bp	Recall	0.7740	0.9100	0.8220	**0.8720**
	Precision	0.7898	0.8198	0.8509	**0.8790**
	F1 Score	0.7818	0.8149	0.8362	**0.8755**
500bp	Recall	0.8420	0.8780	0.8960	**0.9260**
	Precision	0.8287	0.8833	0.9014	**0.9114**
	F1 Score	0.8353	0.8806	0.8987	**0.9186**

Table S4. Comparison of Recalls, Precisions and F1 Scores of the four methods on the CAMI dataset. Bold values represent the best performance.

Criteria	VirFinder	DeepVirFinder	PPR-Meta	Virtifier
Recall	0.3796	0.6522	0.8913	**0.9130**
Precision	0.0249	0.0943	0.2370	**0.2781**
F1 Score	0.0467	0.1648	0.2793	**0.3390**

Table S5. Comparison of Recalls, Precisions and F1 Scores of the four methods on the real human gut metagenomic dataset. Bold values represent the best performance.

Criteria	VirFinder	DeepVirFinder	PPR-Meta	Virtifier
Recall	0.6179	0.7110	0.8738	**0.8937**
Precision	0.6480	0.7589	0.9164	**0.9276**
F1 Score	0.6326	0.7342	0.8946	**0.9103**

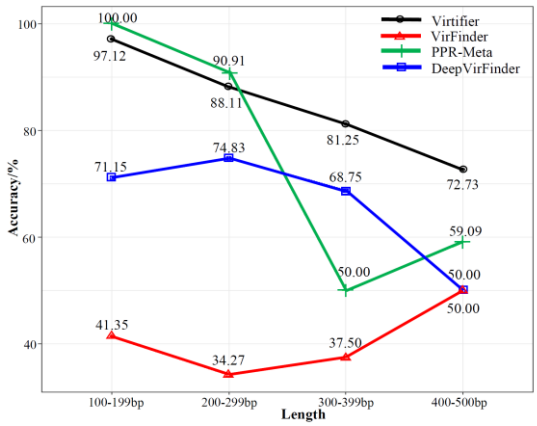
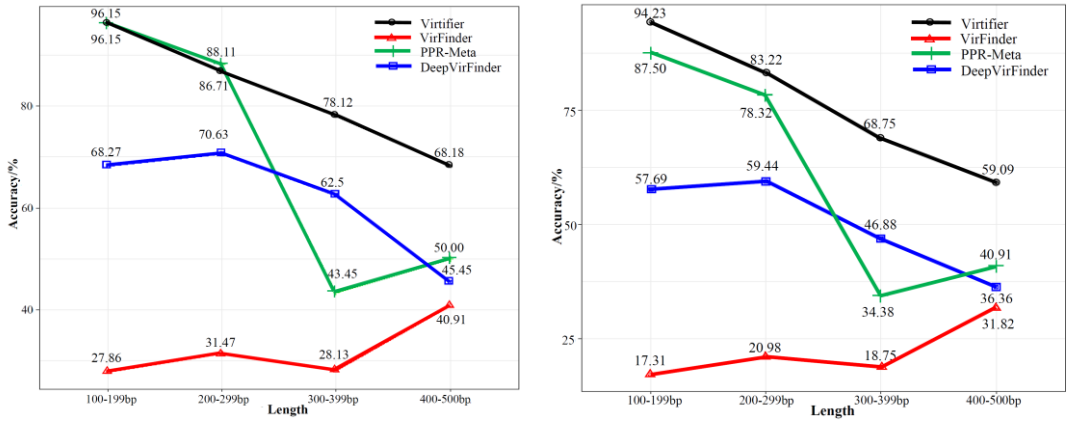
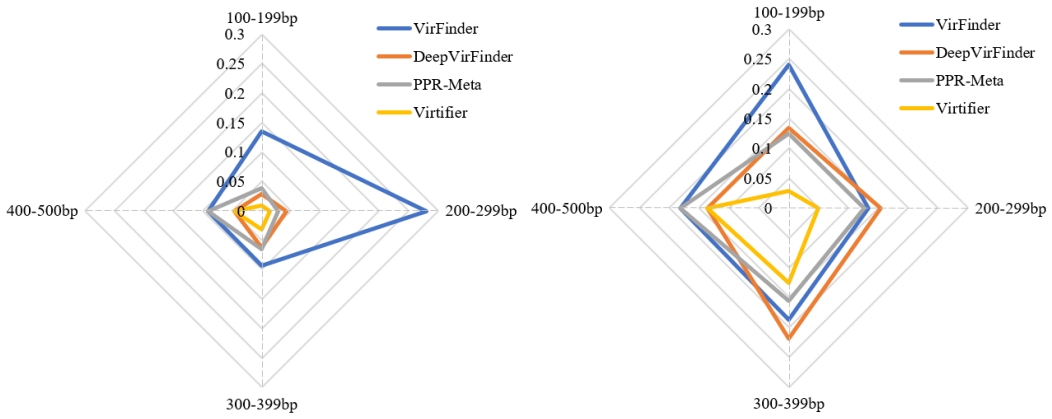


Fig. S7. The proportion of correctly identified viral sequences with different lengths in the real human gut metagenome using the four methods.



(a) base substitutions (b) base insertions or deletions

Fig. S8. The proportion of correctly identified viral sequences in terms of 3% sequencing errors in the real human gut metagenome.



(a) base substitutions (b) base insertions or deletions

Fig. S9. The decrease of the four methods in the situation of 3% sequencing errors in the real human gut metagenome.

S2.5 Comparison results for identifying viral sequences longer than 5,000bp

Table S6. Comparison of four methods on identification viral sequences longer than 5,000bp in real human gut metagenomes. (Num. represents the number of correctly identified viral sequences from all 1270 viral sequences.)

Criteria	VirFinder	DeepVirFinder	PPR-Meta	Virtifier
Num.	981	1,019	**1,096**	1,035
Recall	0.7724	0.8024	**0.8630**	0.8150
Precision	0.7437	0.7814	**0.8424**	0.8333
F1 Score	0.7578	0.7949	**0.8526**	0.8240

The performance of Virtifier on viral identification for long sequences is still better than VirFinder and DeepVirFinder. Virtifier identified 61 less viral sequences than PPR-Meta. In our opinion, the diminishment for Virtifier may be caused by the segmentation of long sequences, which leads to three negative effects. Firstly, when a long sequence was cut off, some useful regions may be separated at the same time. If those regions had relative high weights to contribute to the features of the whole sequence, the operation of cutting off led to losing significant information for identifying viruses. Secondly, there may be relationships between regions in different parts of a long sequence. If the short sequences generated from a long sequence were to be input into Virtifier separately, these relationships would be naturally omitted. Lastly, unbalanced feature distribution in a long sequence led to misclassifying according to the average score from short fragments. Although the precision of Virtifier reduced a little for identifying long sequences, the test performance in the real human gut metagenomic dataset shows that Virtifier still performs better than VirFinder and DeepVirFinder. Although Virtifier identified 4.80% less viral sequences than PPR-Meta, the gap from Virtifier to PPR-Meta for identifying long viral sequences is acceptable.

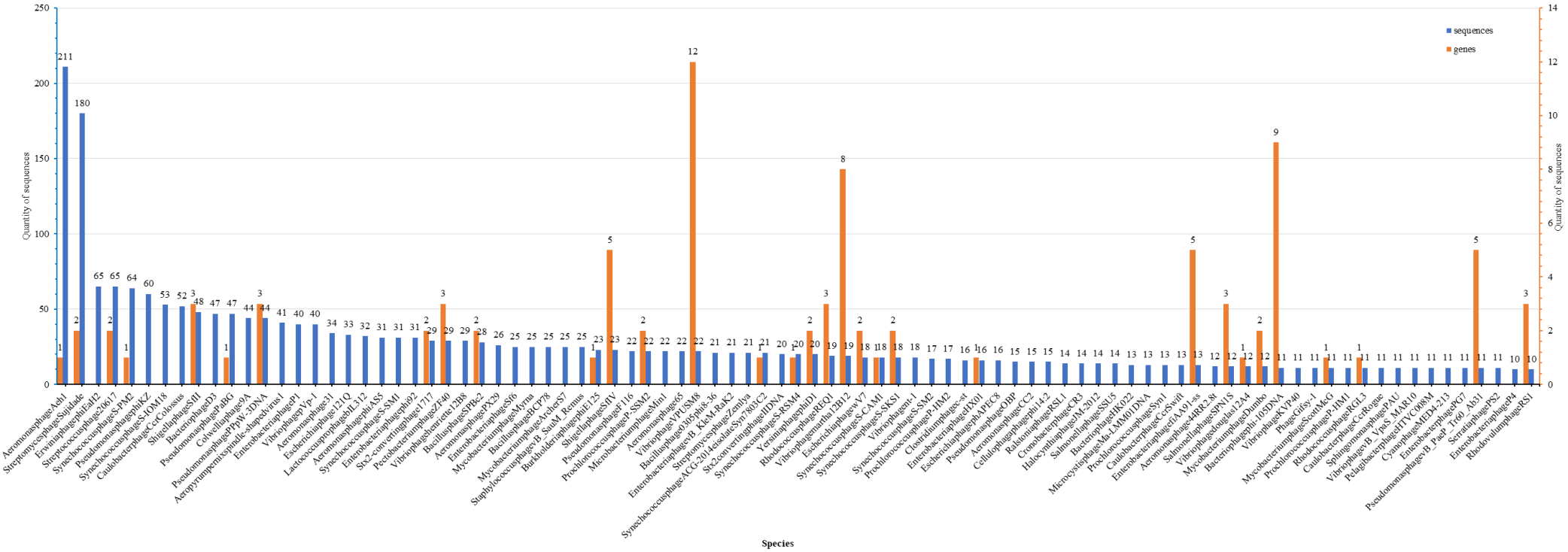


Fig. S10. The quantity of short viral sequences and viral genes being supplemented to each viral species. (89 species with largest quantity of supplemented sequences)

Table S7. The statistical results of BLASTn and CheckV for every viral genome (LS represents the number of long sequences in each viral genome; VGL represents the number of viral genes in long sequences; SS represents the number of short sequences in each viral genome; VGS represents the number of viral genes in short sequences)

Species	LS	VGL	SS	VGS	Species	LS	VGL	SS	VGS
NC_001338.1	6	0	4	0	NC_003387.1	0	0	3	0
NC_001609.1	0	0	10	0	NC_003390.2	0	0	2	0
NC_001697.1	0	0	2	0	NC_003444.1	0	0	6	0
NC_001884.1	6	0	28	2	NC_003524.1	0	0	2	0
NC_001895.1	0	0	2	0	NC_004084.1	5	0	6	0
NC_001900.1	0	0	1	0	NC_004166.2	0	0	3	0
NC_001901.1	0	0	2	0	NC_004167.1	2	1	11	9
NC_001954.1	0	0	1	0	NC_004302.1	0	0	4	0
NC_001956.1	0	0	3	0	NC_004305.1	1	0	1	0
NC_002014.1	0	0	3	0	NC_004313.1	3	7	9	3
NC_002072.2	0	0	1	0	NC_004333.2	2	0	9	0
NC_002166.1	0	0	13	0	NC_004456.1	0	0	2	0
NC_002167.1	0	0	1	0	NC_004466.2	0	0	2	0
NC_002185.1	0	0	1	0	NC_004584.1	3	5	4	0
NC_002214.1	0	0	2	0	NC_004585.1	0	0	2	0
NC_002363.1	0	0	4	0	NC_004586.1	0	0	3	0
NC_002484.2	0	0	47	0	NC_004588.1	0	0	4	0
NC_002486.1	0	0	5	0	NC_004589.1	0	0	3	0
NC_002628.3	0	0	2	0	NC_004629.1	0	0	60	0
NC_002656.1	0	0	1	0	NC_004664.2	3	0	1	0
NC_002667.1	0	0	3	0	NC_004680.1	0	0	1	0
NC_002668.1	0	0	2	0	NC_004682.2	0	0	1	0
NC_002669.1	0	0	1	0	NC_004683.1	0	0	1	0
NC_002670.1	5	0	3	0	NC_004686.2	0	0	1	0
NC_002671.1	14	2	32	0	NC_004688.1	2	0	3	0
NC_002730.1	0	0	1	0	NC_004735.1	0	0	3	0
NC_002747.1	0	0	1	0	NC_004736.1	0	0	1	0
NC_002796.1	0	0	1	0	NC_004745.1	0	0	2	0
NC_003050.2	0	0	3	0	NC_004746.1	0	0	2	0
NC_003157.5	4	0	7	0	NC_004775.2	0	0	2	0
NC_003278.1	0	0	5	0	NC_004813.1	0	0	6	0
NC_003287.2	0	0	1	0	NC_004821.1	0	0	4	0
NC_003291.2	0	0	1	0	NC_004827.1	5	0	7	0
NC_003309.1	2	0	23	1	NC_004913.3	0	0	3	0
NC_003313.1	0	0	1	0	NC_004914.3	2	0	20	0
NC_003315.1	0	0	2	0	NC_004927.1	2	0	4	0
NC_003315.1	0	0	1	0	NC_004928.1	0	0	1	0
NC_003315.1	0	0	2	0	NC_005056.1	0	0	5	0
NC_003315.1	0	0	5	0	NC_005069.1	0	0	2	0
NC_003315.1	0	0	1	0	NC_005083.2	0	0	11	0
NC_003315.1	0	0	2	0	NC_005135.1	0	0	12	0

NC_003324.1	0	0	1	0	NC_005260.1	5	0	211	1
NC_003327.2	0	0	1	0	NC_005262.3	0	0	1	0
NC_003345.1	0	0	1	0	NC_005263.2	0	0	1	0
NC_003356.1	2	3	5	7	NC_005265.1	0	0	2	0
NC_005294.1	0	0	1	0	NC_010179.2	1	0	1	0
NC_005340.1	0	0	1	0	NC_010355.1	1	7	3	1
NC_005342.2	0	0	2	0	NC_010392.1	2	0	11	0
NC_005344.1	0	0	25	0	NC_010393.1	7	75	9	6
NC_005345.2	1	0	5	0	NC_010811.2	10	0	14	0
NC_005357.1	0	0	1	0	NC_011269.2	3	0	11	1
NC_005361.1	0	0	2	0	NC_011273.1	14	3	25	0
NC_005822.1	0	0	1	0	NC_011285.1	2	0	2	0
NC_005841.1	1	0	3	0	NC_011290.1	1	0	5	0
NC_005856.1	0	0	40	0	NC_011357.1	3	0	29	2
NC_005857.1	0	0	3	0	NC_011421.1	2	0	9	0
NC_005879.1	0	0	1	0	NC_011589.1	1	27	1	0
NC_005880.2	1	2	1	0	NC_012635.1	1	0	4	0
NC_005882.1	1	1	6	2	NC_012663.1	2	0	5	0
NC_005885.1	0	0	1	0	NC_012743.2	1	0	1	0
NC_005887.1	0	0	1	0	NC_013055.1	2	5	3	0
NC_005891.1	0	0	1	0	NC_013059.1	1	1	1	0
NC_005948.1	0	0	5	0	NC_013085.1	11	0	20	1
NC_006268.1	1	0	1	0	NC_013588.1	1	0	1	0
NC_006552.1	1	0	22	0	NC_013643.1	1	0	1	1
NC_006820.1	6	0	64	1	NC_013648.1	1	0	3	2
NC_006883.2	13	0	22	2	NC_014036.1	1	0	1	0
NC_006938.1	1	0	2	0	NC_014321.1	5	0	7	0
NC_007021.1	1	0	1	0	NC_014636.1	8	1	31	0
NC_007022.1	10	0	34	0	NC_014900.1	1	0	4	2
NC_007045.1	4	6	2	1	NC_015251.1	7	0	22	0
NC_007051.1	2	1	1	0	NC_015253.1	1	0	5	0
NC_007409.1	4	0	1	0	NC_015254.1	2	0	1	0
NC_007581.1	5	1	16	0	NC_015263.1	4	0	6	0
NC_007805.1	1	1	4	1	NC_015266.1	1	0	2	0
NC_008204.1	1	0	2	0	NC_015269.1	1	0	1	0
NC_008562.1	2	0	13	0	NC_015279.1	5	0	17	0
NC_008584.1	2	0	4	0	NC_015280.1	6	6	11	0
NC_008689.1	3	2	7	0	NC_015281.1	1	0	3	1
NC_008695.1	5	0	1	0	NC_015282.1	6	0	31	0
NC_009237.1	1	0	1	1	NC_015283.1	3	0	6	0
NC_009447.1	1	0	1	0	NC_015284.1	4	0	17	0
NC_009552.2	2	3	5	5	NC_015286.1	3	0	8	0
NC_009603.1	16	0	22	0	NC_015287.1	2	0	5	1
NC_009604.1	1	0	1	0	NC_015288.1	4	0	13	0
NC_009760.1	4	0	21	0	NC_015290.1	11	0	7	0
NC_009762.3	2	0	1	0	NC_015465.1	6	0	3	0

NC_009810.1	1	0	1	0	NC_015466.1	2	0	3	0
NC_009904.1	1	1	6	0	NC_015569.1	2	0	7	1
NC_009986.1	8	0	7	0	NC_016164.1	3	0	1	0
NC_009993.2	3	0	3	1	NC_016564.1	3	0	1	0
NC_016568.1	1	0	1	0	NC_020201.1	1	0	2	1
NC_016571.1	3	2	15	0	NC_020482.1	1	0	4	0
NC_016650.1	4	0	11	1	NC_020484.1	4	0	11	0
NC_016652.1	1	1	2	1	NC_020490.2	1	0	1	0
NC_016654.1	4	4	8	2	NC_020837.1	9	4	18	1
NC_016655.1	2	0	19	3	NC_020839.1	1	0	2	0
NC_016761.1	2	24	12	3	NC_020841.1	1	0	6	0
NC_016766.1	2	1	1	1	NC_020845.1	8	1	11	0
NC_016899.1	7	0	4	0	NC_020851.1	13	0	18	2
NC_017088.1	1	0	1	0	NC_020853.1	3	1	2	0
NC_017974.1	1	0	14	0	NC_020855.1	2	0	7	0
NC_017975.1	1	1	14	0	NC_020859.1	8	0	7	0
NC_018088.1	1	0	44	0	NC_020866.1	1	0	10	3
NC_018279.1	1	0	5	0	NC_020873.1	1	0	3	0
NC_018452.1	1	0	2	0	NC_020874.1	2	1	1	0
NC_018843.1	3	0	14	0	NC_020875.1	3	0	5	1
NC_018848.1	1	0	5	0	NC_021068.1	1	0	12	1
NC_018855.1	1	0	16	1	NC_021070.1	2	3	19	8
NC_018860.1	5	0	25	0	NC_021071.1	1	0	4	0
NC_018861.1	2	2	1	0	NC_021073.1	8	3	29	0
NC_019399.1	1	0	7	0	NC_021299.1	2	0	2	0
NC_019406.1	19	0	52	0	NC_021304.1	4	50	180	2
NC_019408.1	2	0	11	0	NC_021305.1	1	0	1	0
NC_019410.1	1	0	3	0	NC_021306.2	4	30	12	2
NC_019411.1	5	1	13	0	NC_021311.1	1	0	1	1
NC_019445.1	1	0	5	0	NC_021339.1	8	1	21	0
NC_019487.1	1	0	3	0	NC_021346.1	1	0	2	0
NC_019505.1	1	0	18	2	NC_021347.1	2	1	5	0
NC_019516.1	3	0	4	0	NC_021348.1	4	0	25	0
NC_019521.1	1	0	11	0	NC_021529.2	1	0	18	0
NC_019522.1	3	2	29	3	NC_021530.1	1	0	3	0
NC_019526.1	8	0	21	0	NC_021531.1	5	0	8	0
NC_019529.1	9	1	40	0	NC_021534.1	1	0	2	1
NC_019538.1	1	0	15	0	NC_021536.1	16	3	53	0
NC_019543.1	1	0	1	0	NC_021559.1	1	0	2	0
NC_019550.1	1	1	1	0	NC_021560.1	1	0	3	0
NC_019704.1	1	1	1	2	NC_021772.1	1	0	5	0
NC_019713.1	3	2	11	0	NC_021781.2	3	0	6	0
NC_019717.1	1	1	1	0	NC_021794.1	1	0	2	0
NC_019909.1	1	0	1	0	NC_021798.1	3	0	7	0
NC_019917.1	1	0	1	0	NC_021804.1	1	0	4	0
NC_019927.1	1	3	1	1	NC_021806.1	5	0	15	0

NC_019929.1	16	0	65	0	NC_021856.1	5	0	3	0
NC_019934.1	1	0	8	5	NC_021857.1	2	0	48	3
NC_020079.1	2	0	16	0	NC_021861.1	1	12	1	0
NC_020080.1	1	0	3	0	NC_022052.1	3	0	8	0
NC_022058.1	1	0	3	1	NC_023693.1	13	0	31	0
NC_022059.1	2	0	5	0	NC_023697.1	2	0	3	2
NC_022067.1	1	0	6	0	NC_023717.1	3	0	4	0
NC_022071.1	1	0	4	0	NC_023722.1	2	0	2	0
NC_022090.1	1	0	25	0	NC_023724.1	2	0	2	0
NC_022096.1	22	1	47	1	NC_023738.1	1	0	5	0
NC_022747.1	2	2	22	12	NC_023747.1	1	0	5	0
NC_022749.1	2	0	23	5	NC_024121.1	2	0	11	0
NC_022750.1	1	0	13	5	NC_024135.1	1	0	2	0
NC_022914.1	1	0	2	1	NC_024366.1	4	6	3	0
NC_022972.2	1	0	4	1	NC_024792.1	2	0	3	0
NC_022988.1	1	0	2	0	NC_025422.1	2	1	2	0
NC_023006.1	3	0	44	3	NC_025427.1	1	1	2	0
NC_023007.1	2	0	9	0	NC_025429.1	1	0	3	0
NC_023498.1	2	31	3	0	NC_025440.1	1	0	3	0
NC_023502.1	2	0	2	0	NC_025447.1	6	0	33	0
NC_023503.1	21	0	65	2	NC_025455.1	2	0	2	0
NC_023561.1	3	0	11	0	NC_025463.1	3	1	4	1
NC_023564.1	1	0	2	0	NC_026589.1	1	0	1	0
NC_023575.1	1	0	11	5	NC_026594.1	1	0	5	0
NC_023577.1	1	0	2	0	NC_026597.1	2	0	1	0
NC_023581.1	4	1	3	0	NC_026924.1	3	0	5	0
NC_023585.1	1	0	1	0	NC_026926.1	2	0	5	0
NC_023587.1	10	0	2	0	NC_026927.1	3	1	5	0
NC_023591.1	1	0	6	2	NC_026928.1	8	2	21	1
NC_023610.1	2	0	3	0	NC_027353.1	1	0	20	2
NC_023688.1	3	0	26	0	NC_027997.1	1	0	2	0
NC_023690.1	1	0	2	0	NC_028256.1	7	2	7	0
NC_023691.1	1	0	1	0	NC_028268.1	8	0	41	0

S2.6 The necessity of identifying short viral sequences

It is a meaningful advantage to directly identify these short viral sequences because small genome fragments may adversely affect downstream analyses including estimation of viral diversity, host prediction or identification of core genes within viral lineages (Stephen *et al*., 2021). To prove this, the CAMI Marine dataset (https://data.cami-challenge.org/participate) was utilized. Firstly, the assembled contigs from the dataset were separated to a short-contig-subset (<1,000bp) and a long-contig-subset (>1,000bp). These long contigs were input to the DeepVirFinder and these short ones were identified by the Virtifier. DeepVirFinder detected 19,754 viral sequences from all 154,812 long sequences. Virtifier identified 252,960 viral sequences from all

1,233,543 short sequences. Next all of these viral sequences were mapped to virus RefSeqs by BLASTn (E-value $<10^{-5}$) to determine their species information for benchmarking. Viral genes were detected by CheckV (Stephen *et al*., 2021). The results of BLASTn and CheckV are listed in Table S7. 334 viral species were determined from the short-contig-subset (<1,000bp), 248 of which also exist in the long-contig subset. The number of short sequences belonging to the 248 common species and the number of viral genes they carry were counted, as showed in Figure. S10. The short sequences from the metagenome supplement 333 viral sequences and 64 viral genes in total. It is resulted that the short sequences could supplement many viral sequences and genes to the long ones and supplement several species. This proves the necessity of identifying short viral sequences instead of merely identifying sequences after binning.

S2.7 Comparison of running time for Virtifier, VirFinder, DeepVirFinder and PPR-Meta

Table S8. Comparison on Time Consuming of Virtifier, VirFinder, DeepVirFinder and PPR-Meta

Methods		VirFinder	DeepVirFinder	PPR-Meta	Virtifier
Time Consuming (s)	Training	**183**	196	191	346
	Testing	18	18	20	**13**
Maximum Memory (Gb)		**1.6**	2.7	3.3	2.5

We timed Virtifier and the other three benchmark methods (VirFinder, DeepVirFinder and PPR-Meta) in the strategies of training 9,000 contigs of 500bp and testing 1,000 contigs of 500bp from RefSeq genomes. The equipment used for analysis is Intel Core i7-6600U (CPU) with the memory of 8Gb. The time consuming of the two strategies and maximum memory are shown in Table S8. It is worth mentioning that Virtifier, DeepVirFinder and PPR-Meta are all deep learning methods, the running time of which are determined extremely by training iterations. The number of epochs when making a comparison in this paper were all set as 15 (for Virtifier, 15 epochs for the process of training both Seq2Vec and LSTM). Virtifier has the maximum time consumption for training, costing 346 seconds for training 9,000 sequences for 15 epochs and has the minimum time consumption for testing, 13 seconds for testing 1,000 sequences. The huge time consumption of Virtifier during training is caused by the embedding strategy. VirFinder just calculates k-mer frequency before training a machine learning model. DeepVirFinder and PPR-Meta translate bases into one-hot vectors before extracting features. However, Virtifier has to train an extra neural network to learn a meaningful embedding matrix for the bases, which takes a lot of time. At the stage of testing, Virtifier could utilize trained embedding matrix to embed sequences from the testing dataset directly, ignoring the strategy of training embedding matrix, which speeds up the testing procedure. The values of maximum memory during training and testing are similar for the three deep learning-based methods. VirFinder

needs less memory because of there being less parameters in the machine learning model than the deep learning model.

S3 Discussion

Compared to alignment-based methods, although Virtifier is also dependent on reference databases, the LSTM model could learn higher-level features from limited reference sequences in the training set. The identification using these high-level features is different from similarity search in essence. The learned features may represent much more sequences than those currently existing. As a result, Virtifier could correctly identify more viral sequences by the restricted reference database. When it comes to machine learning-based methods, features should be typically designed by human engineers with extensive domain expertise and determining which features are more appropriate for the given task remains difficult. Take VirFinder for example, k-mer frequency as the feature is input into a machine learning model to determine viral sequences. Even though a larger k means a better performance, for short sequences k-mer frequency is sparser and sparser with the increase of k value, which is adverse to identification. Moreover, on comprehensive consideration of time consumption, computational complexity and performance, determining an appropriate k value is tough. Deep learning methods have the ability to learn features automatically. DeepVirFinder and PPR-Meta both utilize CNNs to extract features. However, they feed one-hot encoded sequences to the CNNs. Since the sparsity and mutual independence of one-hot encoding method, these methods lose the co-relationships between every part of a sequence. Virtifier utilize a neural network to learn the high-level representations of a sequence before the feature extracting strategy, which makes up for the lack of information carried by short sequences.

Table S9. Comparison of Recalls, Precisions and F1 Scores of the four models on the testing dataset. Bold values represent the best performance.

Criteria	CNN-single	BOEM	COEM	Virtifier
Recall	0.8640	0.8620	0.8160	**0.9260**
Precision	0.8660	0.8725	0.8020	**0.9114**
F1 Score	0.8625	0.8672	0.8251	**0.9186**

References

Stephen,N. et al. (2021) CheckV assesses the quality and completeness of metagenome-assembled viral genomes. *Nature Biotechnology*, **39**, 578–585.