VirFormer: Transformer On God's Language

Linshu Yang
ShanghaiTech University
yanglsh@shanghaitech.edu.cn

Pengyu Long
ShanghaiTech University
longpy@shanghaitech.edu.cn

Qingcheng Zhao
ShanghaiTech University
zhaoqch1@shanghaitech.edu.cn

***Abstract*—This paper presents a novel deep learning model, VirFormer, for virus nucleotide sequence prediction. VirFormer is a Transformer [1]-based model, which is a powerful sequence modeling tool. We first pretrained an auto-encoder using Transformer based on NCBI virus nucleotide sequences. Then we train a discriminator on the latent space encoded by the encoder to judge whether the sequence is a virus sequence. VirFormer is trained on a large-scale dataset of virus nucleotide sequences. The model is evaluated on a benchmark dataset of virus nucleotide sequences.**

***Index Terms*—bioinformatics, deep learning, virus nucleotide sequence prediction**

I. Introduction

The research of identifying viruses from metagenomics data has attracted much attention in recent years because of the COVID-19 pandemic. Most traditional methods are based on sequence alignment and viral gene databases, which restricts the identification to existing viruses in databases and thus cannot detect new viruses. Since the viruses keep evolving, the traditional methods are not reliable enough for virus prevention and control. Some alignment-free algorithms have been developed to resolve this problem, such as VirFinder [2] and DeepVirFinder [3] which is the enhanced version of VirFinder with deep learning. The longer the sequences are, the better the DeepVirFinder performs. Some other methods aim to improve the accuracy when sequences are short, such as Virtifier [4] and VirSorter2 [5].

However, the performance of these methods is still not satisfactory. The reason is that the existing methods are not able to capture the long-range dependencies in the sequences. The Transformer [1] is a powerful sequence modeling tool, which can capture the long-range dependencies in the sequences. Besides, the existing methods are based on labeled data, which is not enough for training a deep learning model. We proposed an auto-encoder to learn the latent space of virus sequences, which can be trained on unlabeled data.

II. Background

There are several methods that can be used to identify viral DNAs from a collection of DNA sequences, including sequence similarity searches, and machine learning approaches.

But using sequence similarity searches needs a lot of human labeling and a huge databases. These approaches rely on the availability of a database of known viral DNA sequences. The databases should be large enough and only the known virus can be found in the databases. If the viral DNA in the sample is significantly different from known viral DNA sequences, it may not be identified by the search algorithm. Another potential disadvantage of sequence similarity searches is that it can be computationally intensive, especially if the database is large. Although we have many fast alignment approach like BLAST (Basic Local Alignment Search Tool) to compare the DNA sequences in question to a database of known viral DNA sequences, it is still difficult to perform these searches in a timely manner, considering searching through a whole database.

Machine learning approaches can be used to classify sequences as viral or non-viral based on features extracted from the sequences. These features might include the presence of certain motifs or patterns in the DNA, the GC content of the sequence, or the presence of certain amino acids in the encoded proteins. VirFinder [2] is a tool for identifying viruses from metagenomics data. It is based on a k-mers counting method. VirFinder first extracts the k-mers from the sequences, then counts the k-mers in the sequences. After encoding the sequences, VirFinder uses a logistic regression model to classify the sequences. It is trained on NCBI RefSeq dataset, and has a AUROC score of 0.87 on the benchmark dataset. It works well when the sequences are long, but performs poorly when the sequences are short.

Based on a convolutional neural network, DeepVirFinder [3] is the enhanced version of VirFinder with deep learning. DeepVirFinder first encoded the sequences using Motif detectors, then uses a convolutional neural network to classify the sequences. The prediction results are outputted by a binary classifier. It is trained on NCBI RefSeq dataset, to further extend the dataset, researchers also included a large amount of metagenomic data of viruses. It has a AUROC score of about 0.90 on the benchmark dataset, which also affected by the length of the sequences.DeepVirFinder also presents a method to divide training set, validation set and test set by time series to validate that the model is effective on new virus.

Using more domain knowledges, VirSorter2 [5] is a multi-classifier to detect diverse DNA and RNA viruses guided by expertises. It trains 5 separate random forest classifiers on different dataset (customized by major type of viral groups) to classify the sequences. It is also trained on NCBI RefSeq dataset. To process the sequences, VirSorter2 uses a sliding window on the whole sequence to acquire significant sequence.

Virtifier [4] is a deep learning-based identifier for viral sequences from metagenomes. It uses a Seq2Vec model to encode the sequences and trains a LSTM model and an attention layer to decode. It is also trained on NCBI RefSeq dataset. The attention mechanism can capture the features of the sequences effectively and the pretrained encoder also works well on the sequences of different lengths.

The existing methods show a trend that the model and methods from NLP are more and more applied to bioinformatics (Virtifier [4], AlphaFold2 [6], DNABert [11], scBERT [10]). However, the performance of existing methods in finding viruses from metagenomics data is not satisfactory enough (VirFinder [2], DeepVirFinder [3], Virtifier [4], VirSorter2 [5]). Regarding the task of protein structure prediction as a task harder than a classification of virus, AlphaFold2 [6] can perform well on protein structure prediction with confidence of more than 95% [6]. We consider the reason of impressive performance is that AlphaFold2 is a attention-based model, which can perform well on sequence processing tasks.

DeepVirFinder shows that more data from metagenomics can improve the performance of the model. However, labeling the data is a time-consuming task and needs a lot of human resources and professional knowledge. The idea of using a pretrained encoder from Virtifier give us a hint that we can use a pretrained encoder to learn the latent space of virus sequences. So we decided to train an auto-encoder on metagenomics data from NCBI, the data may include virus sequences and non-virus sequences to enhance the performance of the encoder.

Nucleotide sequences are sometimes with the size of thousands of base-pairs, which is a challenge for the model to process. The Virtifier [4] uses LSTM to cope with this problem. However, the LSTM is still hard to capture the long-range dependencies in the sequences [12].

To overcome this issue, we propose the VirFormer, a model architecture avoiding using large numbers of labeled viral DNA sequence data and relying on an attention mechanism to synthesize DNA representations, and then classify the sequences as viral or non-viral. Using Transformer as backbone, which based on encoder-decoder model and multi-head attention, the VirFormer focuses on different positions of the input sequence to compute a representation of that sequence, which we expected it to be some biological pattern of the nucleotide sequence, including the presence of certain motifs or DNA patterns or the presence of certain amino acids in the encoded proteins. With the enhancement of attention mechanism, our model can capture information about the entire sequence, enabling it to learn long-range dependencies. This helps extracting more features, representing higher level structure of folded sequences. The VirFormer uses a discriminator to judge whether the sequence is a virus sequence, learning from the latent space of sequences synthesized by the encoder.

III. Methods

Our goal is to train a Transformer-based autoencoder which can learn the feature of DNA sequence and a classifier which can verify whether a sequence is from a virus. The idea of training a Transformer-based autoencoder is mainly supported by 2 reasons.

First is the self-attention mechanism. Self-attention is a mechanism used in deep learning models to enable the model to focus on certain parts of the input when processing it. It allows the model to selectively weight the importance of different parts of the input, rather than treating all parts of the input equally. Self-attention is often used in natural language processing tasks, such as machine translation and language modeling, where it can help the model to better understand the relationships between words in a sentence and their importance in relation to one another. Since there exists non-coding DNAs which will not take part in the protein synthesis, we consider that different parts in a DNA sequence may have different weight in determining the trait of organisms. Also, noticing that proportion of non-coding DNAs in DNA sequence can vary greatly between different organisms and coding DNAs may distribute discretely in a whole DNA sequence. Such features give up inspiration that encoding DNA sequence may have some similarity to NLP task. Based on the facts mentioned above, it may be difficult for a traditional encoder based on linear or CNN layer to handle the latent feature of DNA sequence. Hence, we consider the self-attention mechanism as a better approach to learn a latent space from DNA sequence.

Second is that an autoencoder support unsupervised learning, which enable us to train our model on a larger amount of data. One challenge in the task of virus identification is that the data is unbalanced, which means that the number of virus sequences is much smaller than the number of non-virus sequences due to the limitation of sequence length of virus. Therefore, it is enough for the model to handle the features of virus sequence. If we want the autoencoder to learn the feature of both virus and non-virus data, we should ensure that virus data takes a certain proportion in the whole data. However, the size of virus data is limited(only 160MB in all) so that the size of the whole dataset will also be strickly limited. Thus, we use anomaly detection mechanism that training the autoencoder on a large amount of non-virus data and consider virus as an anomaly, which is a unsupervised learning task and eschewed the unbalanced data problem. Anomaly detection, also known as outlier detection, is the process of identifying unusual or unexpected patterns in a dataset that do not conform to the expected behavior. Anomaly detection is used in a variety of applications, including fraud detection, intrusion detection, and system monitoring. We train the autoencoder on the whole NCBI RefSeq dataset without virus data including about 160GB data consisting of DNA sequence from prokaryotes and human. In our model, the virus sequence will be regarded as an anomaly since the autoencoder is trained on a large amount of non-virus data.

After the auto-encoder is fully trained, we fix the parameters of the encoder and train a discriminator on the latent space encoded by the encoder. The discriminator is a binary classifier to judge whether the sequence is a virus sequence. According to the anomaly detection mechanism, the classifier can capture the virus sequence as an anomaly.

A. Modeling Nonsupervised DNA Latent Space

1) Autoencoder Architecture: The architecture of a transformer-based autoencoder consists of two main components: the encoder and the decoder. The encoder takes in the input data and compresses it into a latent representation, while the decoder takes the latent representation and reconstructs the original input data.

The encoder and decoder both consist of multiple layers of self-attention and feedforward neural network blocks. The self-attention blocks allow the transformer to capture long-range dependencies in the input data and the feedforward blocks allow it to learn more complex patterns.

The transformer architecture also includes a multi-head attention mechanism, which allows the model to attend to multiple parts of the input data at the same time. This allows the transformer to capture more nuanced patterns in the data and improve the quality of the latent representation.

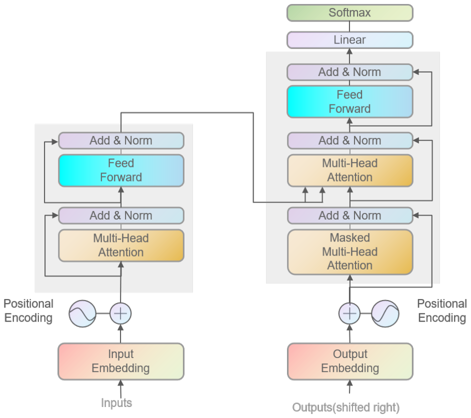

Fig. 1. Network Architecture, Autoencoder
https://github.com/dair-ai/ml-visuals

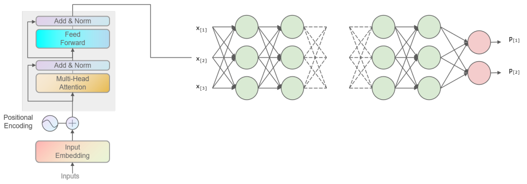

Fig. 2. Network Architecture, discriminator
https://github.com/dair-ai/ml-visuals

2) Loss of Autoencoder: In the training phase of the autoencoder, the model is training based on a customized NLL loss. Comparing to the original NLL loss, we add a KL divergence loss [16] to the original NLL loss to encourage the latent space to be a normal distribution.

$$\mathcal{L} = \mathcal{L}_{NLL} + \mathcal{L}_{KL}$$

where

$$\mathcal{L}_{NLL} = -\frac{1}{N}\sum_{i=1}^{N} logP(x_i)$$

$$\mathcal{L}_{KL} = \mu^2 + \sigma^2 - log(\sigma^2) - \frac{1}{2}$$

where N is the number of sequences in the batch, x_i is the i-th sequence in the batch, μ and σ are the mean and standard deviation of the latent space, respectively. Since we consider the virus sequence verifying task as a anomaly detection task, we add KL divergence to the loss such that the latent space of non-virus sequences will be under the normal distribution. When the input is a virus DNA sequence, the distribution of the latent space may also changed, which enable the classifier to judge whether the sequence is a virus sequence more easily.

B. Learning the DNA Latent Space

We implemented a CNN model with a bottle-neck structure and less complex linear layers.

To be more specific, we increased convolution layers from 2 to 4. We use LeakyRelu as activation function.

In addition, to help our model to stablize, we added spectral normalization [17] to our convolution layer by forcing the model to be Lipschitz continuous.

IV. Experiments and Results

A. Experimental Settings

1) Dataset: We use 90% of human DNA sequence (GRCh38_latest_genomic, about 34GB) and DNA sequence of prokaryotes before 2015 (about 12GB) to train the autoencoder and test it on the rest human DNA (about 4GB) and prokaryotes DNA after 2015 (about 2GB).

We will use non-virus data from NCBI RefSeq dataset to train the autoencoder. Using non-virus data to train the autoencoder can also enlarge the amount of data we can use. The sequence will be clipped into 250bp length since we lay more attention on the ability of the model on short sequence. Each nucleotide bases (adenine (A), cytosine (C), guanine (G), thymine (T) and N) will be encoded into one-hot vector before being feeded into the autoencoder, where N represents for a placeholder for a nucleotide base that is not known or cannot be determined. Since N nucleotide bases rarely appears(about 3 in 100,000 bases), it makes little difference on the reconstruction results.

When training the classifier, virus sequence before 2015 will be used to train and validate the model, and virus sequence after 2015 will be used to test the model in order to test whether the model is effective on new virus. The proportion of virus and non-virus data is 10:1. However, to avoid the bias of the dataset, we will use the balanced sampling techniques to create a more balanced training set. Sequences will have different probability p_i to be sampled, which is determined by the length of the sequence and whether it is viral. The longer the sequence is, the higher probability it will be sampled. Then at each epoch, the number of samples of each sequence is determined by the probability of the sequence. The number of samples of viral sequences obeies the binomial distribution $Bin(n, p_i)$, where n is the number of expecting sample number of the dataset.

B. Training

The training process has two phases: pre-training the autoencoder and fine-tuning the classifier.

We first preprocess the input data, which includes clipping the sequence into 250bp length and encoding the nucleotide bases into one-hot vector. Then we train the autoencoder for 40 epochs with a batch size of 20. We train the autoencoder on eight Tesla V100 GPUs, which takes about 2 hours to train one epoch. The model weights are initialized randomly. And the random seed is set to a constant for reproducibility all through the experiments. Then, the preprocessed input data is fed through the encoder and decoder to produce a reconstructed output. The NLL loss is calculated as the difference between the input data and the reconstructed output. The model weights are updated using Adam optimization algorithm [13], to minimize the reconstruction error.

C. Evaluation

In this section, we show that our model successfully learns a latent representation. Then we evaluate the performance of the model on the task of virus sequence verification. For the evaluation, we use the AUROC and accuracy as the metrics. In the ablation study, we calculate the accuracy, AUROC score and F1 score of different autoencoders to test whether KL divergence loss and anomaly detection works.

We also compared the performance of the autoencoder trained by full data(including virus) and non-virus data, with KL divergence loss and without KL divergence loss respectively. The result shows that both the idea of adding KL divergence to the loss function and regarding virus as an anomaly works.

1) Representativeness and Informativeness of DNA Latent Space: We first train the autoencoder with the original NLL loss function, the negative log loss of the predicted probability is converged at aprroximately 10^{-4}, which shows that the Transformer model can learn the latent space of sequences and reconstruct the sequences.

2) Ablation Study on KL Loss and Anomaly Detection: Then we add the KL divergence to the loss function. We compared the performance of the autoencoder trained by full data(including virus) and non-virus data, with KL divergence loss and without KL divergence loss respectively. The result shows that both the idea of adding KL divergence to the loss function and regarding virus as an anomaly improve the performance of the model. To evaluate the performance, the classifiers for these autoencoders are kept the same.

Results of the ablation study is shown in Table. **??**, where the accuracy, AUROC score and F1 score of different autoencoders are shown.

Since we keep the same dataset as the DeepVirFinder, we can compare the performance with it.

3) Classifier Evaluation: The architecture of the classifier are compare between linear model, naitve CNN and our CNN classifier.

The linear model is consist of 4 linear layer with the size of $(2048, 512, 128, 1)$.

The naitve CNN is consist of a convolutional layer and a linear layer. The convolutional layer does not use spectral normalization and LeakyRelu.

TABLE I
ABLATION STUDY

	Accuracy	AUROC Score	F1 Score
Fulldata w.o. KL	89.40	0.97	89.61
Non-virus w.o. KL	93.24	0.98	93.37
Non-virus with KL	92.20	0.98	94.53
DeepVirFinder	-	0.95	-

TABLE II
ABLATION STUDY

	Accuracy	AUROC Score	F1 Score
Linear	70.79	0.77	93.37
Naitve CNN	86.97	0.94	85.73
Ours	93.24	0.98	69.46
DeepVirFinder	-	0.95	-

V. CONCLUSION

We proposed an autoencoder to learn the latent space of sequences, which can be trained on unlabeled data. The model is based on a Transformer, which can capture the long-range dependencies in the sequences. Based on the autoencoder, a discriminator can judge whether the sequence is a virus sequence. The encoder is pretrained on NCBI RefSeq dataset without virus data, which can be trained on unlabeled data. Our model shows a better performance than DeepVirFinder on the benchmark dataset with sequence length of 500bp.

In the future, we would like to explore other directions to further improve our framework.

First, we can use k-mer to represent the sequences. In the context of DNA sequencing, a k-mer is a contiguous sequence of k nucleotide bases in a DNA molecule. they provide a way to summarize the sequence information in a compact form, which can help the autoencoder capture more latent features from the DNA sequence.

Second, we can use a more powerful discriminator to judge whether the sequence is a virus sequence.

Third, we can also train our model on different sequence length. Up to now, we have only test our model on 500bp DNA sequence since we have really limited time to do the research. Comparing VirFormer with other related works on various DNA sequence length can provide more information for further improving our model.

To solve the problem that self-attention mechanism takes a lot of time and memories to process the long sequences, we can also use a linear attention mechanism to replace the original self-attention mechanism to reduce the time and memory complexity from $O(n^2)$ to $O(n)$. For example, Performer [14] or Nyströmformer [15].

To sum up, Transformer model applied on DNA sequence verifying is a promising direction. Due to the limitation of time and computing resources, we only tried a simple model based on transformer. We believe that based on the Transformer model, we can achieve better performance on DNA sequence verifying.

REFERENCES

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser. Attention Is All You Need. arXiv:1706.03762v5 [cs.CL] 6 Dec 2017.

[2] Ren, J., Ahlgren, N.A., Lu, Y.Y. et al. VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data. Microbiome 5, 69 (2017). https://doi.org/10.1186/s40168-017-0283-5

[3] Ren, J., Song, K., Deng, C. et al. Identifying viruses from metagenomic data using deep learning. Quantitative Biology, 8, 64-77 (2020). https://doi.org/10.1007/s40484-019-0187-4

[4] Miao Y, Liu F, Hou T, Liu Y. Virtifier: A deep learning-based identifier for viral sequences from metagenomes. Bioinformatics. 2021 Dec 15:btab845. doi: 10.1093/bioinformatics/btab845. Epub ahead of print. PMID: 34908121.

[5] Guo, J., Bolduc, B., Zayed, A.A. et al. VirSorter2: a multi-classifier, expert-guided approach to detect diverse DNA and RNA viruses. Microbiome 9, 37 (2021). https://doi.org/10.1186/s40168-020-00990-y.

[6] Jumper, J., Evans, R., Pritzel, A. et al. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583-589, 2021. https://doi.org/10.1038/s41586-021-03819-2.

[7] Jason M Norman, Scott A Handley, Megan T Baldridge, Lindsay Droit, Catherine Y Liu, Brian C Keller, Amal Kambal, Cynthia L Monaco, Guoyan Zhao, Phillip Fleshner, et al. Disease-specific alterations in the enteric virome in inflammatory bowel disease. Cell, 160(3):4470460, 2015.

[8] Alejandro Reyes, Laura V Blanton, Song Cao, Guoyan Zhao, Mark Manary, Indi Trehan, Michelle I Smith, David Wang, Herbert W Virgin, Forest Rohwer, and others. Gut DNA viromes of Malawian twins discordant for severe acute malnutrition. Proceedings of the National Academy of Sciences, 112(38): 11941-11946, 2015.

[9] Samuel Minot, Rohini Sinha, Jun Chen, Hongzhe Li, Sue A Keilbaugh, Gary DWu, James D Lewis, and Frederic D Bushman. The human gut virome: inter-individual variation and dynamic response to diet. Genome research, 21(10):1616-1625, 2011.

[10] Yang, F., Wang, W., Wang, F. et al. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. Nat Mach Intell 4, 852-866 (2022). https://doi.org/10.1038/s42256-022-00534-z

[11] Yanrong Ji, Zhihan Zhou, Han Liu, Ramana V Davuluri, DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome, Bioinformatics, Volume 37, Issue 15, 1 August 2021, Pages 2112-2120, https://doi.org/10.1093/bioinformatics/btab083

[12] Vaswani, Ashish, et al. ”Attention is all you need.” Advances in neural information processing systems 30 (2017).

[13] Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. doi:10.48550/ARXIV.1412.6980

[14] Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., ... & Weller, A. (2020). Rethinking attention with performers. arXiv preprint arXiv:2009.14794.

[15] Xiong Y, Zeng Z, Chakraborty R, Tan M, Fung G, Li Y, Singh V. Nyströmformer: A Nystöm-based Algorithm for Approximating Self-Attention. Proc Conf AAAI Artif Intell. 2021;35(16):14138-14148. Epub 2021 May 18. PMID: 34745767; PMCID: PMC8570649.

[16] Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. doi:10.48550/ARXIV.1312.6114

[17] Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957.