ViraMiner-CB: Deep Learning Architecture Inspired from GoogleLeNet for Robust Identification of Viral DNA Sequences

Federico Pivotto, Fabrizio Genilotti, Leonardo Egidati

University of Padua

Abstract

Despite their clinical relevance, identifying highly divergent or entirely new viruses poses a significant challenge. During the sequencing of human samples, standard alignments often categorize numerous assembled contigs as "unknown" because these sequences do not match any known genomes. In literature, one of the proposed approaches in the field of deep learning is ViraMiner, a convolutional neural network designed for the identification of viruses in human biological samples. ViraMiner employs two neural network branches, developed to detect both the patterns and the frequencies of viral patterns in raw metagenomic contigs. Our study propose a new branch that implements an enhancement of the strategies utilized in the branches of ViraMiner. The new branch modifies the first convolutional layer to simultaneously extract diverse types of information by employing two different convolutional operations: the first applies a 1D convolution and the second a 2D convolution. This innovation is inspired by the architecture of GoogleLeNet, which represents the first implementation and application of the Inception Layer. We suggest that the developed model can effectively capture diverse types of genetic information and thereby enhance the performance of the previously proposed ViraMiner network.

Introduction

The human virome encompasses all viruses present within the human body. These viruses vary significantly between healthy and ill individuals. Despite their clinical significance, the full impact of viruses on health is not yet fully understood, and their detection and classification pose significant challenges. Metagenomic studies have identified numerous new viruses, suggesting that many human viruses remain undiscovered. There are also indications that certain unidentified viruses may be involved in the onset of autoimmune diseases such as diabetes and multiple sclerosis.

To tackle the detection of viral DNA sequences, one method employs machine learning techniques to learn from examples and classify the presence of viral genomes, generalizing to new human samples. Among various machine learning models published for virus detection in metagenomic data, ViraMiner utilizes Convolutional Neural Networks (CNN) on raw metagenomic contigs to identify potential viral sequences in diverse human samples. ViraMiner's structure extends the basic architecture of CNN in order to enhance its effectiveness in addressing this specific classification problem. In this study, we have expanded the architecture of ViraMiner by developing two network configurations based on the Inception Layer, a concept introduced with the GoogleLeNet network by Christian Szegedy et al. (Szegedy et al. 2014). The modifications implemented in ViraMiner introduce a new custom branch that retains the structure of the original branches but replaces part of the 1D convolutional layers with an innovative component inspired by the Inception Layer. This layer allows to perform two different operations in parallel to capture diverse information on the input. The first configuration of ViraMiner integrates the pattern branch with the custom branch tuned for analyzing frequencies. Conversely, the second configuration pairs the frequency branch with the custom branch tuned for analyzing patterns. These extensions are designed to enhance the model's classification strategy, effectively addressing the complexity of viral data in human biological samples For training the models, we used the same dataset used by the authors of ViraMiner, namely 19 metagenomic experiments from sample types such as skin, serum, and condylomas. For this purpose, DNA one-hot encoding method is used to represent DNA sequences as numerical data that can be processed by generic machine learning algorithms.

These models achieve significantly improved accuracy compared to other existing methods for identifying viruses in metagenomic samples.

Requirements

Convolutional Block

The major modification implemented on the custom branch is the introduction of a Convolutional Block inspired by the Inception Layer. This block consists of two sub-layers operating in parallel on the same input: the first performs a 1D convolution, while the second performs a 2D convolution, as depicted in Figure 1. Both 1D and 2D convolution operations use kernels with the attributes outlined in Table 1, so as to allow the concatenation of the outputs of the two sub-layers.

Exploiting the convolution operation in two dimensions, it is possible to extract relevant information of the local structure of a DNA viral sequence, which could be useful to find

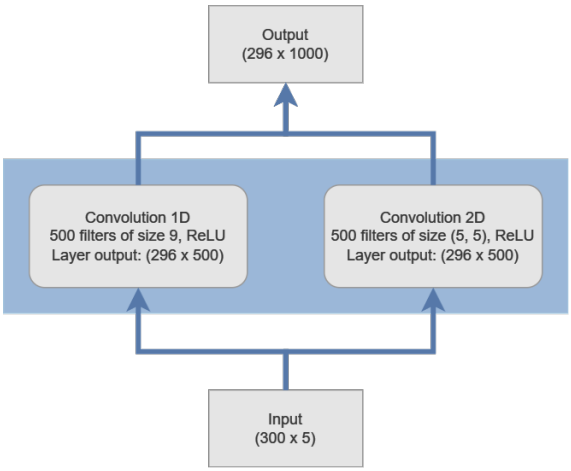

Figure 1: Illustration of the Convolutional Block structure for the custom frequency branch.

patterns that may be overlooked by the original branches. Moreover the block is lighter than the Inception Layer since to extract relevant information from sequences are need few convolutional operations.

Table 1: Kernel characteristics of the custom sub-layers

Kernel	Custom branch
1D Convolution	Size = (9,1), Padding = 2, Stride = 1
2D Convolution	Size = (5,5), Padding = 0, Stride = 1

Network Architectures

In our work, we started from the ViraMiner convolutional neural network presented by Ardi Tampuu et al. (Tampuu et al. 2019) to develop a new approach based on the Convolutional Block described, which we integrated into a new network branch. Considering this custom branch and the ViraMiner network, we built two different configurations by pairing the proposed branch with the branches of ViraMiner, namely the frequency branch and the pattern branch. Both the configurations take as input a DNA sequence and output the confidence score of detecting a viral sample.

ViraMiner-CB

The new branch is designed to complement the existing architecture of ViraMiner, providing a more robust feature extraction mechanism. Although in the custom branch the layers, except the first, are not modified, the Convolutional Block is a fundamental change in the structure with respect to the branches of ViraMiner. As described in the previous section, the Convolutional Block implements two parallel operations, useful to extract more complex patterns. In order to introduce the branch in the two configurations, we have developed two versions of it: one capturing the frequency in data, the second the patterns in the sequence (Pivotto, Genilotti, and Egidati 2024). In Figure 2 the custom branch tuned for frequency is highlighted in the blue box.

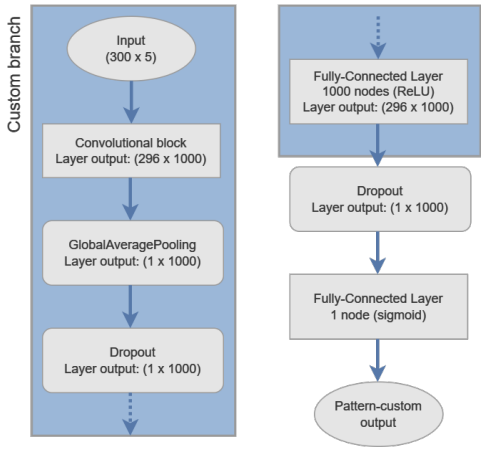

Figure 2: Architecture of the customised pattern branch in frequency-custom configuration.

Table 2: Branches performance on test set with respect to the viral class.

Custom branches	Precision	Recall
custom-frequency	0.75	0.17
custom-pattern	0.55	0.33

Proposed configurations Here we introduce the two configurations to enhance the architecture of ViraMiner. Both are based on a dual-branch convolutional neural network architecture where two CNNs run concurrently on the same input, with their outputs merged through concatenation before the last layer. The first configuration is composed by the pattern branch of ViraMiner and the custom branch tuned for capturing the frequency information on the patterns. The second one is composed by the frequency branch of ViraMiner and the custom branch built for capturing the pattern information on the patterns. To avoid an increasing complexity and inference time of the configurations, the custom branch computes the convolutions using half the number of filters for each sub-layer inside the Convolutional Block.

Materials and Methods

Training and Validation

Training the custom branches Before starting the training phase of the final configurations, the single custom branches are trained to analyze their discriminative capabilities. The training process is made of 10 epochs with validation made on learning rate, learning rate schedule and dropout probability. The chosen loss function to be minimised is the binary cross-entropy.
The performances of the custom branches are reported in Table 2. According to the recall and precision scores, it can be seen that the custom branch for frequency information classifies better the samples of the non-viral class (less false-positives), while the one for pattern information the viral samples (less false-negatives).

Table 3: Performance of model configurations on test set with respect to the viral class.

Model configuration	AUROC	Precision	Recall
frequency-custom	0.918	0.83	0.25
pattern-custom	0.922	0.84	0.32

Training the configurations In order to perform the training and validation of the architecture, we used the same dataset adopted by ViraMiner authors, obtained from 19 metagenomic experiments. For the training process, we considered the binary cross-entropy loss. Considering the hyperparameters values tested in ViraMiner, we performed validation searching for the best values of dropout probability and maximum learning rate for our model configurations. During training, the model was fit on the dataset running 15 epochs, using the cosine annealing learning rate schedule that refresh every 5 epochs, and randomly initialising the weights.

Networks Comparison

Once the two configurations pattern-custom and frequency-custom were trained, we have computed the evaluation metrics, in particular considering the test-accuracy, the confusion matrix and the ROC curve. Using these evaluation metrics, it is possible to consistently compare our approach with ViraMiner, in particular we consider the precision and recall of the viral class to understand the capability of the models to handle viral sequences. As stated by the authors of ViraMiner, its best model provides a ROC curve with an area under the curve of 0.924, and achieves 90% precision and 32% recall on the viral class on test set. The novel models of our work have performances reported in Table 3. It can be seen that overall the pattern-custom model is the best among the two configurations, achieving a ROC curve score of 0.922, while the frequency-custom model has a ROC curve score of 0.918. The best configuration achieves a precision of 84% and a recall of 32% which has a decrease in precision score but same recall score compared with ViraMiner performances.

Results

Network Performance

Among the considered configurations, the pattern-custom model outperforms the frequency-custom one, suggesting that the two branches of pattern-custom are able to capture different features of the input data and provide a more comprehensive representation of the data respect to the frequency-custom model. Indeed, the pattern branch is capable to better recognise the viral class, while the custom branch tuned for capturing frequency information can classify more correctly the non-viral class. Overall the novel approach proposed in this work built a model with a lower performance compared to ViraMiner, in particular by looking at the precision score of 90% in Viraminer while 84% in pattern-custom. The reported results could be attributed to several factors. First, the two branches in both the configurations were not pre-trained, thus not having that advantage in training phase. Secondly, the Convolutional Block contains as many feature maps as the ViraMiner branches, therefore it might be beneficial to either increase the number of feature maps or maintain the current number while redistributing them across the two sub-layers. Finally, the Convolutional Block comprises only two sub-layers. Increasing the number of sub-layers, whether 2D or 1D, might help capture relationships between more distant nucleotides. It could be interesting to refine this new approach in this way to see whether it is possible to reach higher performances and discriminate better viral and non-viral samples. In Figure 3 is reported the ROC curve of the pattern-custom configuration.

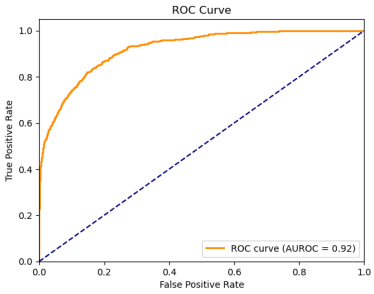

Figure 3: Plot of the ROC curve of pattern-custom configuration

Conclusion

This work proposes a novel approach for the viral vs non-viral sequences binary classification task, which was addressed by the authors of ViraMiner. A new architecture inspired from ViraMiner is presented in two configurations, the pattern-custom and the frequency-custom. The two configurations make use of a new layer called Convolutional Block, which is made of two convolutional sub-layers: one performing the convolution in 1D, the other operating in 2D. In both configurations only the custom branch implements the Convolutional Block, which is made on purpose to analyse the effect of the branch in the architecture.
Overall the pattern-custom model is the best of the two configurations, however it has lower precision with respect to ViraMiner since it scores 84% against the 90% of ViraMiner. This result could be improved by refining the training strategy, exploring more in detail the configurations of the architecture, and improving the Convolutional Block structure.
Once the architecture is refined, a future project could involve the implementation of a new configuration using the Convolutional Block in both branches of the configuration to better exploit the potential of the network in discriminating viral and non-viral DNA sequences.

References

Pivotto, F.; Genilotti, F.; and Egidati, L. 2024. ViraMiner-CB. https://github.com/FedericoPivotto/viraminer-cb.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2014. Going Deeper with Convolutions. arXiv:1409.4842.

Tampuu, A.; Bzhalava, Z.; Dillner, J.; and Vicente, R. 2019. ViraMiner: Deep Learning on Raw DNA Sequences for Identifying Viral Genomes in Human Samples.