# Residual Neural Network in Genomics

Sara Sabba, Meroua Smara, Mehdi Benhacine, Loubna Terra, Zine Eddine Terra

**Abstract**

Residual neural network (ResNet) is a Deep Learning model introduced by He et al. in 2015 to enhance traditional convolutional neural networks proposed to solve computer vision problems. It uses skip connections over some layer blocks to avoid vanishing gradient problem. Currently, many researches are focused to test and prove the efficiency of the ResNet on different domains such as genomics. In fact, the study of human genomes provides important information on the detection of diseases and their best treatments. Therefore, most of the scientists opted for bioinformatics solutions to get results in a reasonable time.

In this paper, our interest is to show the effectiveness of the ResNet model on genomics. For that, we propose two new ResNet models to enhance the results of two genomic problems previously resolved by CNN models. The obtained results are very promising and they proved the performance of our ResNet models compared to the CNN models.

**Keywords:** Deep Learning, genomics, convolutional neural network, companion, Residual neural network, super-enhancers, viral genomes.

## 1 Introduction

Machine Learning (ML) is one of the artificial intelligence fields, which is interested in the design and development of intelligent algorithms that learn and evolve with experiences to discover knowledge or make decisions (predictions) without being humanly guided or explicitly programmed to handle particular data. Indeed, the learning process begins with observations of data, experience, instructions, or examples to find the best model which can be able to make the best decisions in the future.


doi:10.56415/csjm.v30.17

Deep Learning (DL) is the emerging technique of Machine Learning. Its basic concepts and models have derived from the Artificial Neural Network which mimics the activity of the nervous system of the human brain to intelligentize algorithms and avoid tedious human labor. DL has several computational models such as Deep fully connected neural networks (DNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Auto-encoder, Generative adversarial networks (GAN), Graph convolutional neural networks (GCN), Residual Neural Network (ResNet), etc. Most of them have provided their effectiveness in specific research areas such as computer vision [51, 52], natural language processing [53, 54, 55], and signal processing [56].

Currently, Deep Learning is an extremely active research area in bioinformatics [7, 15, 24, 26, 27, 37] due to the massive evolution of biological data. Its algorithms proved their efficiency in many critical life situations. They allow predicting many diseases, treatments, and biological phenomena from the analysis and interpretation of various types of data [1, 9, 10, 22, 24, 41]. In fact, most of the bioinformatics research is focused on Molecular biology which usually is called genomic. It is mainly interested in studying the cell at the molecular level, i.e., understanding the interactions between the different molecular systems of a cell, including the interactions between these macromolecules (DNA, RNA, and protein biosynthesis), as well as learning how these interactions are regulated.

In fact, many bioinformatics frameworks based on Deep Learning were developed in the literature to solve genomics problems. Xu et al. [37] proposed DeepEnhancer framework for predicting enhancers using convolutional neural networks (CNN). They used the FANTOM5 permissive enhancer dataset, JASPAR database and ENCODE cell type-specific enhancer datasets to train and test their model. Zhou et al. [40] developed a Deep Learning-based algorithmic framework, called DeepSEA to predict the noncoding-variant effects de novo from sequence. The proposed model is trained and tested on a regulatory sequence code from large-scale chromatin-profiling data. Alipanahi et al. [3] used also deep convolutional neural networks to develop DeepBind approach for predicting the sequence specificities of DNA- and RNA-

binding proteins. This approach is trained and tested on in vitro data, and it addressed many challenges we cite: (i) it can be applied to both microarray and sequencing data; (ii) it can tolerate a moderate degree of noise and mislabeled training data and (iii) it can train predictive models fully automatically, alleviating the need for careful and time-consuming hand-tuning. Likewise, SpliceFinder [34] and Splice2Deep [2] were designed to predict splice sites of human genomic using CNN model. The both works are trained and validated on some genomic sequences such us Homo sapiens, Oryza sativa japonica, Musmusculus, Drosophila melanogaster, and Daniorerio. In fact, there are so many critical frameworks worthy of our interest that we cannot cite them all.

Most of the genomic Deep Learning solutions are based on the CNN models. As is already known, CNNs are very useful in solving image classification and visual recognition problems. However, studies have shown that if the network has too many layers, we can observe the degradation of performance due to the vanishing or exploding gradient problem [57]. Accordingly, ResNet was introduced [42] to solve this problem by using skip connections (identity connections) or shortcuts (that create residual blocs) to jump over some layers which allow the network to retain what it has previously learned.

Recently, researchers are motivated to implement new genomic solutions using ResNet models we cite: Li et al. [44] developed ResPRE method to predict residue-level protein contacts using inverse covariance matrix of multiple sequence alignments. Sun et al. [45] proposed RNAcontact algorithm for predicting RNA inter-nucleotide 3D closeness. Shuvo et al. [46] introduced QDeep method to present new distance-based single-model quality estimation by harnessing the power of stacked deep ResNets. Zhang et al. [47] predicted Gene Expression from DNA Sequence. Zhang and Shen [49] proposed ThreaderAI to improve protein tertiary structure prediction. Kandel at al. [50] presented PUResNet model for predicting protein-ligand binding sites. Li and Xu [48] developed a new model of convolutional residual neural network for predicting protein structure using Inter-residue distance prediction. Wang et al. [43] proposed RPRes to predict RNA secondary structure profile. However, the number of these proposals remains modest compared to the CNN ones.

In this paper, we propose two new residual neural network models for two genomic problems. The first proposition aims for predicting super-enhancers on a genome scale, and the second aims for predicting viral genomes. Our purpose is to improve the results obtained by previous solutions based on CNN models [5, 59] and prove the effectiveness of ResNet models in genomic science. Moreover, there are three reasons behind this motivation: (i) first, ResNet was created to optimize the performance of CNN for avoiding the vanishing gradient problem, (ii) second, to the best of our knowledge, none of the literature research on super-enhancers or vital genome prediction is utilizing ResNet-based approach, and (iii) third, the obtained results proved the performance of our proposals compared to the CNN models.

# 2 Related works

## 2.1 Super-enhancers prediction

The prediction of super-enhancers (SEs) has prominent roles in biological and pathological processes. They play critical roles in the control of cell-type-specific genes programs, especially that related to the detection and progression of tumors [8, 14, 18, 32, 36]. SEs are defined as clusters of transcriptional enhancers. They are formed by binding of high levels of enhancer-associated chromatin features that drive high-level expression of genes encoding key regulators of cell identity [16, 26, 30].

The identification of SEs is based on the differences in their ability to bind markers of promoter transcriptional activity [32], including cofactors such as mediators (MED1, MED12) and cohesions (Nipbl, Smc1), histone modification markers (H3K27ac, H3K4me1, H3K4me3, H3K9me3), chromatin regulators (Brg1, Brd4, Chd7), and chromatin molecules (p300, CBP). Furthermore, Whyte et al. [35] indicated five embryonic stem cell (ESC) transcription factors to occupy super-enhancers (Oct4, Sox2, Nanog, Klf4, and Esrrb). However, there are many additional transcription factors that contribute to the control of ESCs [27, 29, 39]. In [14], authors tested ChIP-Seq data for fifteen additional transcription factors in ESCs and explored whether they occupy enhancers defined by Oct4, Sox2, and Nanog (OSN) co-occupancy. Their experiment results showed that six additional transcription fac-

tors (Nr5a2, Prdm14, Tcfcp2l1, Smad3, Stat3, and Tcf3) occupy both typical enhancers and super-enhancers and that all of them are enriched in super-enhancers.

Recently, many studies [23, 31, 32] proved that gene transcriptional dysregulation is one of the core tenets of cancer development that involves in noncoding regulatory elements, such as TFs, promoters, enhancers, SEs, and RNA polymerase II (Pol II). In particular, SEs play core roles in promoting oncogenic transcription to accelerate cancer development [4, 7, 32]. Recent research showed that cancer cells acquire super-enhancers at the oncogene, and cancerous phenotype relies on these abnormal transcription propelled by SEs [13, 25]. Accordingly, it is important to understand super-enhancers and their components since they control much disease-associated sequence variation that occurs in these regulatory elements [12, 14, 21] in large amounts of data in order to better understand biological processes. This knowledge can lead to discoveries that improve quality of life (i.e., designing more effective medical treatments or discovering certain severe illness in its early stages).

There are few bioinformatics works based on Machine Learning proposed to predict super-enhancers of the genomes. Authors of [19] implemented and compared six different Machine Learning models to identify key features of SEs and to investigate their relative contribution to the prediction. The six models include: Random Forest, Support Vector Machine, k-Nearest Neighbor, Adaptive Boosting, Naive Bayes, and Decision Tree. To validate their idea, they used 10-fold stratified cross-validation, independent datasets in four human cell-types and a set of publicly available data. Authors of [5] proposed a new computational method called DEEPSEN for predicting super-enhancers based on a convolutional neural network. The proposed method is trained and tested on 36 SEs features, where 32 ones are used in [19], and 4 others are selected from ChIP-seq and DNase-seq datasets.

## 2.2 Viral genomes prediction

Viral metagenomics is the science that studies human, animal, and plant viral diseases. It consists of describing the total viral genome, or

virome for the discovery of new viruses. The results of metagenomics have allowed advances in diagnosis, molecular epidemiology, and viral evolution, and these studies have great relevance for re-evaluating concepts in pathology and, in particular, the biological role of viruses in an organism [60, 61, 62].

The detection of potential viral genomes in human biological samples is a crucial step in the viral metagenomics process. It currently represents an interesting problem in the field of bioinformatics. It aims to identify a human virome in DNA sequences extracted by a previous phase of metagenomics. Indeed, viruses are reservoirs and carriers of genes, suggesting that the human virome may have played a central role in human adaptation and evolution [64]. This importance reveals the need to update the methods used by this science.

In fact, there are some bioinformatics works based on Machine Learning proposed to identify viral DNA sequences. Ren et al. [64] implemented VirFinder based on the k-mers approach. Ren et al. [65] proposed a new computational method called DeepVirFinder based on a convolutional neural network. Tampuu et al. [59] enhanced the previous approach by proposing a parallel model called ViraMiner which is based on two CNN branches configured differently.

# 3 Proposed models

This section is devided into two parts. The first one presents the ResNet model proposed to predict super-enhancers, and the second part describes the ResNet model proposed to identify viral genomes.

## 3.1 ResSEN: Residual Neural Network for predicting super-enhancers

### 3.1.1 Datasets

The public database used to train and test our approach is used in [19] and [5] published previously. In fact, there are 36 features (see Table 1) incorporated in publicly available ChIP-seq and DNase-seq datasets of mouse embryonic stem cells (mESC) taken from Gene Expression Omnibus (GEO).

Table 1. Features of datasets used by [5] and our approach.

| Super-enhancers data type | Features |
|---|---|
| Histone modifications | H3K27ac, H3K4me1, H3K4me3, H3K9me3 |
| DNA hypersensitive site | DNaseI |
| RNA polymeraseII | Pol II |
| Transcriptional co-activating proteins | p300, CBP |
| P-TFEb subunit | Cdk9 |
| Sub-units of Mediator complex | Med12, Cdk8 |
| Chromatin regulators | Brg1, Brd4 and Chd7 |
| Cohesin | Smc1, Nipbl |
| Subunits of Lsd1-NuRD complex | Lsd1, Mi2b |
| Histone deacetylase | H-DAC2, HDAC |
| Transcription factors | Oct4, Sox2, Nanog, Esrrb, Klf4, Tcfcp2l1, Prdm14, Nr5a2, Smad3, Stat3, Tcf3 |
| Sequence signatures | AT content, GC content, phastCons, phastConsP, repeat fraction |

The datasets contain 11100 samples. Among them, 1119 are positive and 9981 are negative. To train, test, and compare our ResSEN approach, we divided those samples into training datasets and test datasets, where 90% (i.e., 9990) are used for training and 10% (i.e., 1110) are used for performance testing (see Table 2).

Table 2. Division of samples.

| Datasets | Samples size | Positive samples | Negative samples |
|---|---|---|---|
| Training datasets | 9990 | 1006 | 8984 |
| Test datasets | 1110 | 113 | 997 |

Notice that the samples used in the validation phase are the same used in the test phase because the total number of samples is insufficient to be devised into three sub datasets.

### 3.1.2 ResSEN model

ResSEN model is composed of an input layer, a convolution layer, a pooling layer, two residual blocks and a fully connected layer.

***a. Input layer***

Thirty six (36) characteristics are used to predict the super-enhancers (see Table 1). So, there are 36 nodes in the input layer. The values of these nodes are normalized (using Eq. (1)) and standardized (using Eq. (2)) before they are transmitted to the next network layers.

$$y = (x - min) + (max - min), \tag{1}$$

where $x$ is the input node value, and $max, min$ are the maximum, minimum values between input nodes.

$$z = (y\text{–}mean)/standard\_deviation, \tag{2}$$

where $y$ is the normalized node value, $mean$ is calculated using Eq. (3), and $standard\_deviation$ is calculated using Eq. (4):

$$mean = sum(y)/count(y), \tag{3}$$

$$standard\_deviation = \sqrt{(sum((y - mean)^2)/count(x))}. \quad (4)$$

***b. Convolutional layers***

ResSEN is composed of five convolutional layers: i) a convolutional layer before the first residual block, and ii) two convolutional layers for each residual block ($2 \times 2 = 4$).

In the first convolutional layer we applied 64 filters of size $1 \times 7$, followed by Max-pooling with pool-size $1 \times 3$ and stride 1. The first residual block has two convolutional layers, we applied 128 filters of size $1 \times 3$ in the first one, and 256 filters of the same size $1 \times 3$ in the second one. The second residual block has also two convolutional layers. In the first layer, we applied 256 filters of size $1 \times 3$, while in the second layer, we applied 512 filters of the same size $1 \times 3$.

Figure 3 illustrates the filters' parameters of the five convolutional layers.

Each convolutional layer is followed by a Batch Normalization (BN) layer (see Fig. 1) which is used to improve the speed, performance, and stability of deep neural networks [11], [17].

***c. Activation layer***

Deep learning usually employs a multilayer network and the gradient algorithm to train models, therefore it requires heavy computing, and the learning is often trapped into local minima. Currently, studies propose the rectified linear unit (ReLU) as the activation function to address this problem because its gradient is simple to compute, which allows the model to train easier, faster, and perform better [7].

Consequently, ResSEN uses ReLU as an activation function:

$$ReLU(x) = max(0, x). \quad (5)$$

***d. Add identity***

For each residual block, ResSEN uses convolution block strategy to add the block's input to the block's output. This type of design requires that the block's output and its input have the same shape (size), so they can be added together. The output of the first block will be the

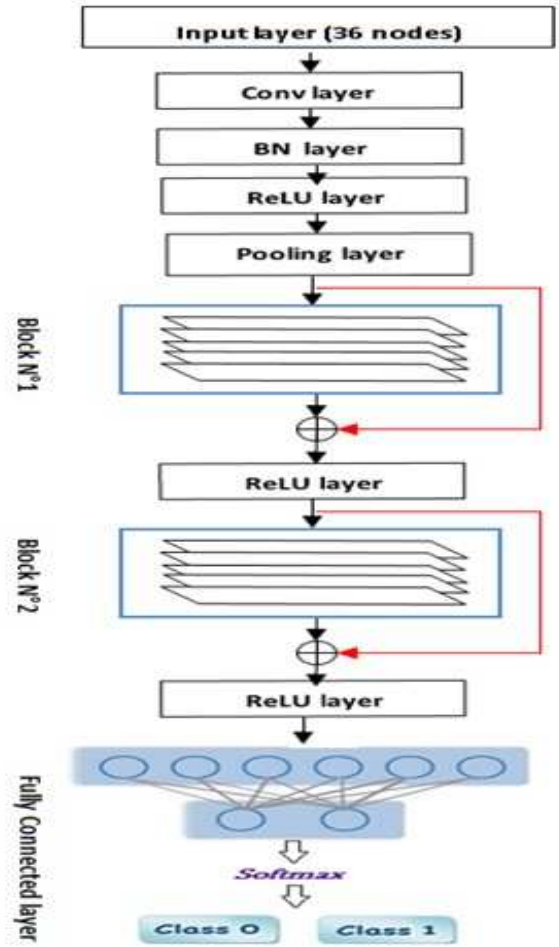



Figure 1. ResSEN model

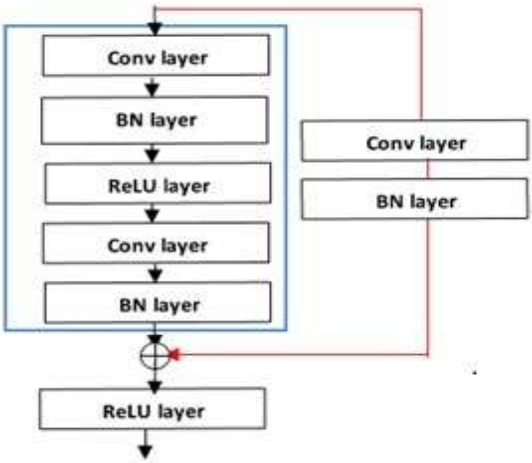



Figure 2. Residual Block of ResSEN

input of the second block and the output of the second block will be the input of the fully connected layer. The structure of each residual block is shown in Fig. 2.

To transform the block's input into the desired shape, we introduced 256 convolutions (256 filters) of size 1×3 for the first residual block and 512 convolutions (512 filters) of size 1×3 for the second residual block (see Fig. 3).

***e. Fully connected layer***

The fully connected (FC) layer of the ResSEN is structured as follows:

- The number of input neurons is 17408.
- The activation function is ReLU.
- The number of output layer is 2 neurons.
- The function used to calculate the probability of the output classes is: Softmax (see Eq. (6)).

$$Softmax(x_j) = max \frac{e^{x_i}}{\sum_j e^{x_i}}, j \in \{1, 2, ....k\}, \tag{6}$$

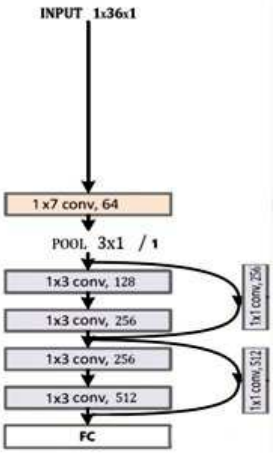



Figure 3. ResSEN convolutional layers parameters

where, $k$ is the number of classes. Moreover, to obtain the predicted class A, we applied the $argmax$ function to the $Softmax$ function output:

$$A = argmax(Softmax(x_j)). \tag{7}$$

- So, if $A = 1$, the predicted class is positive, which means the presence of the super-enhancers in the genome;
- if $A = 0$, the predicted class is negative, which means the absence of super-amplifiers in the genome.

### 3.1.3 ResSEN training

ResSEN training is based on supervised learning, which consists of calculating the optimal weights using the input matrix D (the data samples) and the output matrix A (the desired outputs or the class labels) corresponding to D. D is a matrix of size $N \times 36$, and $A$ is a binary matrix of size $N \times 1$, where $N$ is the number of samples, which is set to 9900. $A[i] = 1$ if the corresponding sample represents

the super-enhancer class, otherwise, $A[i] = 0$. During the training phase, ResSEN uses the cross entropy loss function that measures the difference between the calculated output and the desired output (see Eq. (8)).

$$J(\Theta) = \frac{1}{n}\sum_{i=1}^{n} a^i log(h_\Theta(x^i)) + (1 - a^i) log(1 - h_\Theta(x^i)), \qquad (8)$$

where, $\Theta$ is the set of parameters, $n$ is the number of samples, $a^i$ is the label of $x^i$, $h_\Theta(x^i)$ is the predicted label of $x^i$.

To update ResSEN weights, we used Backpropagation model and Adam method [20]. The latter is an adaptive learning rate optimization algorithm that is designed to improve the classical method of stochastic gradient descent (SGD) aiming at accelerating deep neural network learning. It automatically adapts the learning rate for each parameter by calculating adaptive estimates of moments.

## 3.2 ResVG: Residual Neural Network for predicting viral genomes

### 3.2.1 Datasets

The datasets used to train, validate, and test our approach are used in the work of Tampuu et al. [59] published previously. There is a set of metagenomic sequences taken from different samples such as skin, serum, and condyloma, obtained by merging and mixing 19 experiments. These last are divided into medium-sized ones (of 300 bp). The datasets contain 264049 samples. Among them, 5551 are positive (viral sequences) and 258498 are negative (non-viral sequences). To train, test, and compare our ResVG approach, we divided those samples into training, validation, and test datasets, where 80% (i.e., 211239) are used for training, 10% (i.e., 26405) are used for validation, and 10% (i.e., 26405) are used for performance testing (see Table 3).

### 3.2.2 ResVG model

ResVG model is composed of an input vector, convolutional layers, batch normalization layers, ReLU activation layers, a Max pooling

Table 3. Division of viral genome samples

| Datasets | Samples size | Positive samples | Negative samples |
|---|---|---|---|
| Training datasets | 211239 | 4466 | 206773 |
| Test datasets | 26405 | 551 | 25854 |
| Validation datasets | 26405 | 534 | 25871 |

layer, a residual block, a Global Average Pooling layer, and a fully connected layer.

***a. Input data***

ResVG uses DNA sequences of length 300 bp, each one is coded in binary on 4 bits. Indeed, there are 4 possible values (ACGT): A = 1000, C = 0100, G= 0010, and T = 0001. So, each input vector corresponds to a 1D sequence (an array) of length 300 with 4 channels (as shown in Fig. 4). This means that there are 1200 input neurons for the network.

***b. Convolutional layers***

ResVG is composed of three convolutional layers: i) a convolutional layer before the first residual block; ii) two convolutional layers for the residual block.

In the first convolutional layer, we applied 64 filters of size $1 \times 7$, followed by Max-pooling with pool-size $1 \times 4$ and stride 1. The residual block has two convolutional layers; for both, we applied 1000 filters of size $1 \times 11$.

As the first proposal, each convolutional layer is followed by a Batch Normalization layer (BN) and ReLU (rectified linear unit) activation layer.

***c. Add identity***

For a residual block, ResVG uses also convolution block strategy to add the block's input to the block's output. Therefore, to transform the ResVG block's input into the desired shape, we introduced 1000 convolutions (1000 filters) of size $1 \times 11$ (see Fig. 4).

***d. Fully connected layer***

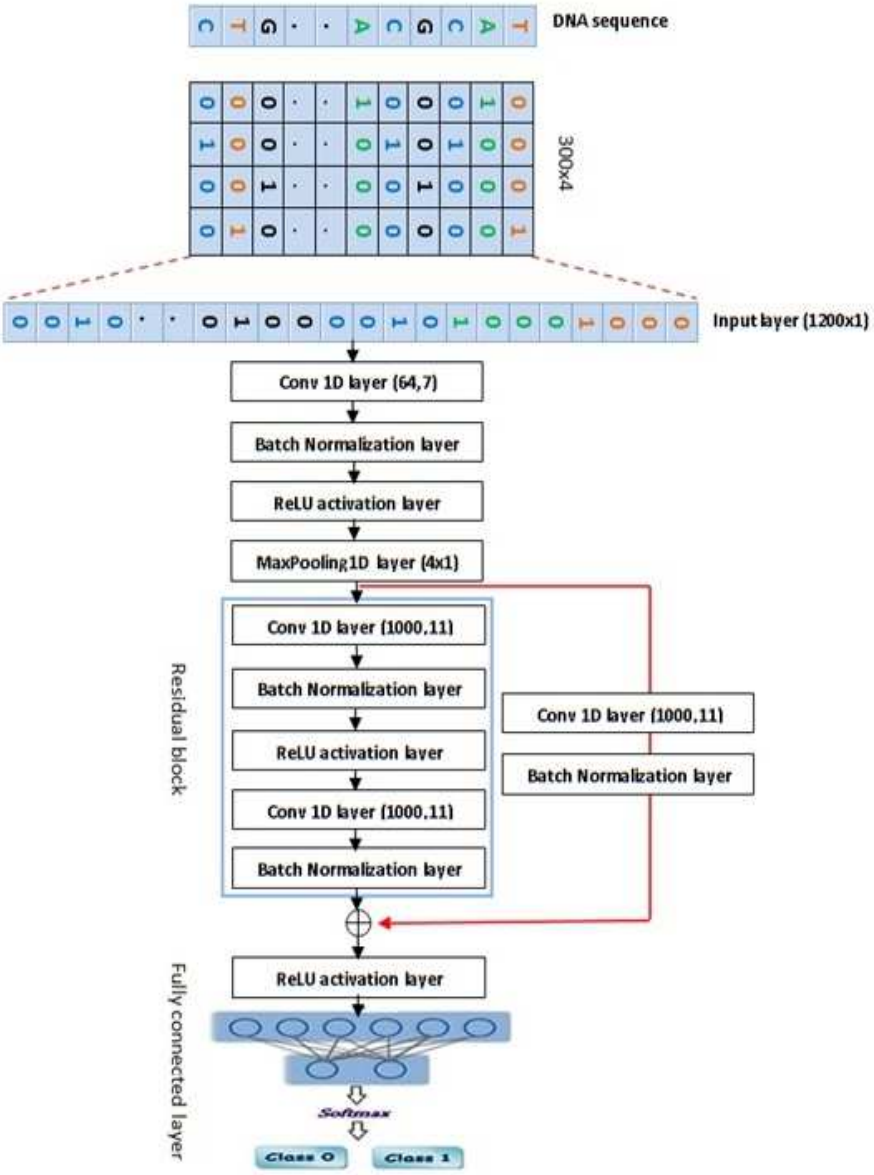



Figure 4. ResVG model

The fully connected (FC) layer of the ResVG is structured as follows:

- The number of input neurons is 1001.
- The activation function is ReLU.
- The number of output layer is 2 neurons.
- The function used to calculate the probability of the output classes is: Softmax (see Eq. (6)).

#### 3.2.3 ResVG training

ResVG training is based on supervised learning, which consists of calculating the optimal weights using the input matrix $D$ (the data samples) and the output matrix $A$ (the desired outputs or the class labels) corresponding to $D$. $D$ is a matrix of size $N \times 1200$ ($300 \times 4$), and $A$ is a binary matrix of size $N \times 1$, where $N$ is the number of samples which is set to 211239. $A[i] = 1$ if the corresponding sample represents the viral class, otherwise, $A[i] = 0$.

During the training phase, ResVG uses the cross entropy loss function to measure the difference between the calculated output and the desired output, and Backpropagation model and Adam method [22] to update network weight's.

## 4 Experimental results and comparison

In the context of binary classification, the evaluation of models is based on some performance measures that are computed from the confusion matrix (see Table 4). Thus, to evaluate and compare our model's performance with those published by DeepSEN [5] and ViraMiner [59], we calculated accuracy, recall, precision, and F1-score for ResSEN model and accuracy, precision, and AUROC (TPR vs FPR) for ResVG model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{9}$$

$$TPR(TruePositiveRate)/Recall = \frac{TP}{TP + FN}, \tag{10}$$

$$Precision = \frac{TP}{TP + FN}, \tag{11}$$

Table 4. Confusion matrix

| | | Actual class | |
|---|---|---|---|
| | | + | - |
| Predicted class | + | **T**rue **P**ositives | **T**rue **N**egatives |
| | - | **F**alse **P**ositives | **F**alse **N**egatives |

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}, \tag{12}$$

$$FPR(FalsePositiveRate) = \frac{FP}{TN + FP}. \tag{13}$$

Notice that the TPR and FPR are used by the AUROC curve to represent the separability degree between classes.

The best results obtained by testing the best models of ResSEN and ResVG are compared respectively with those of DeepSEN and ViraMiner. Those comparisons are shown in Figs. 5 and 6.

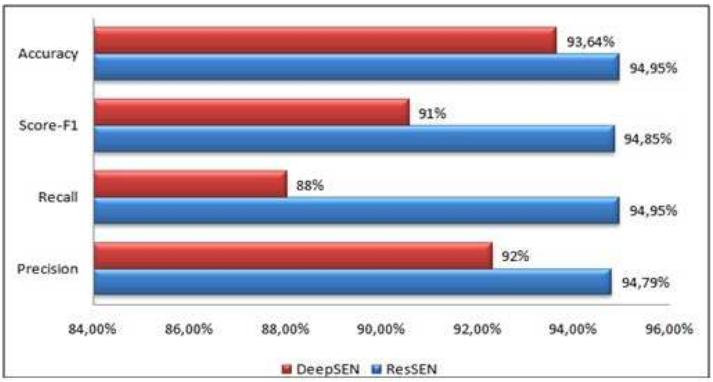
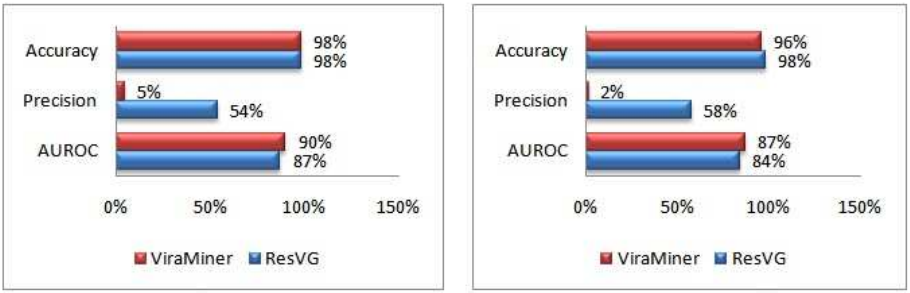



Figure 5. Performance comparison graph of ResSEN and DeepSEN in the validation and the test phases

In the DeepSEN paper, the authors proposed a model with three convolutional layers (followed each one by a pooling layer) and a fully connected layer. They mentioned that their model achieved an accuracy of 98% [5]. However, by checking the DeepSEN code published




(a). Performance comparison of ResVG and ViraMiner in the test phase.

(b). Performance comparison of ResVG and ViraMiner in the validation phase.

Figure 6. ResVG results

in [6], we found that they used 80% of samples in the training phase and all datasets (100% of samples) in the testing phase, which is wrong according to the learning strategy [33], [38]. Ideally, the model should be tested on samples that were not used in the training phase.

So, to ensure a fair comparison with our ResSEN model, we re-executed the DeepSEN using 90% of samples for training and 10% of samples for testing. The obtained results (see Fig. 8), show that the best model of the DeepSEN achieves an accuracy of 93,64% and a precision of 90%. However, in both cases (validation with all the datasets or with 10% of samples), we noticed the presence of the overfitting problem. The latter is clearly modeled in the accuracy and loss curves that we generated after re-executed DeepSEN (see Fig. 7). Knowing that, the blue and the orange curves represent the development of accuracy/loss in the training phase and in the validation phase, respectively.

Figure 8 shows the accuracy and loss curves of ResSEN model. In this case, there is no overfitting problem. We noticed a harmonization between the curves generated in training and test phases. Finally, the results shown in Figs. 5, 7, and 8 prove that our proposed model outperforms that of DeepSEN for the prediction of super-enhancers.

Furthermore, in ViraMiner paper, authors proposed a model with two branches. The first uses a single convolutional layer of 1000 filters followed by GlobalMaxPooling layer. The second branch also uses

a single convolutional layer of 1200 filters followed by GlobalAverage-Pooling layer. The results of the two branches are concatenated to find the inputs of the last FC layer.

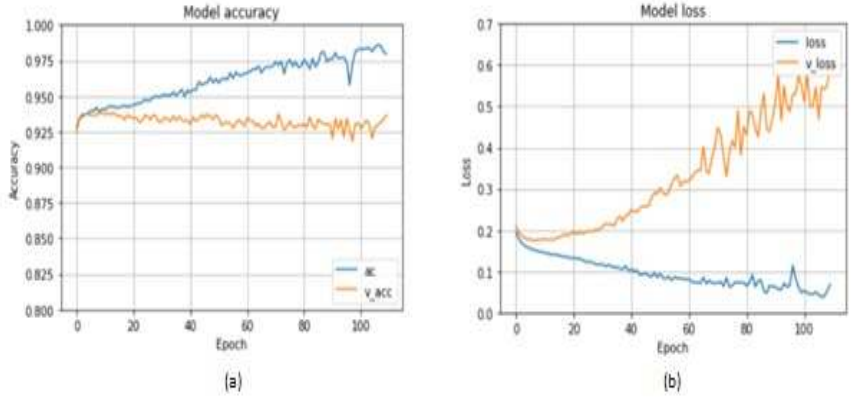



Figure 7. (a) DeepSEN accuracy curve, (b) DeepSEN loss curve

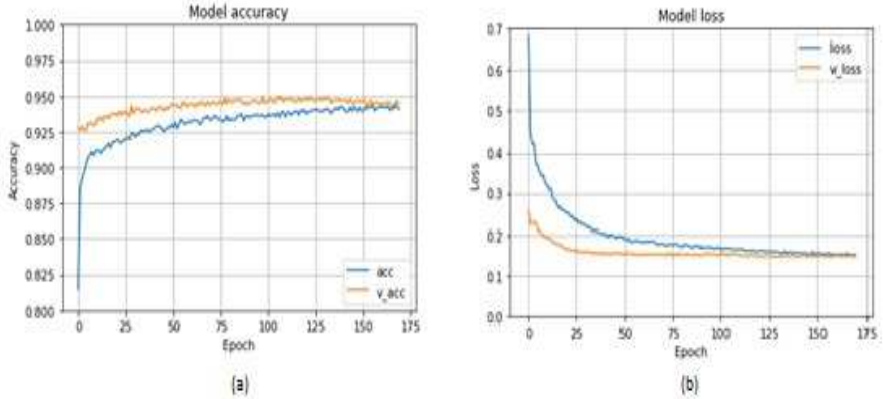



Figure 8. The feature vector set

To ensure a fair comparison with our ResVG model, we re-executed the ViraMiner model published in [66] using 80% of the samples for training, 10% for validation, and 10% for testing. The obtained results are shown in Figs. 6, 9, and 10.

Comparing the results of ViraMiner with our results, we noticed

that the AUROC curve of ViraMiner is the best; however, there is a big difference in the precision performance. Moreover, by analyzing the loss curves, we noticed that the ViraMiner model has a big overfitting problem compared to our Model. Finally, we can say that our ResVG model has optimized the prediction performance of viral genomes compared to the ViraMiner model, especially in the validation phase.

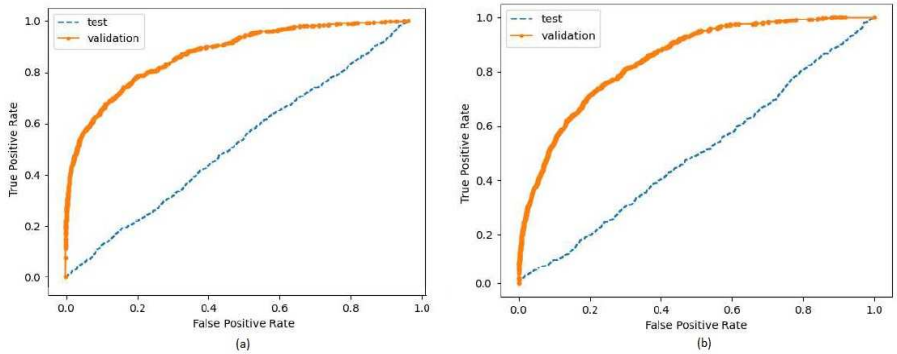



Figure 9. (a) The AUROC curve of ViraMiner, (b) The AUROC curve of ResVG

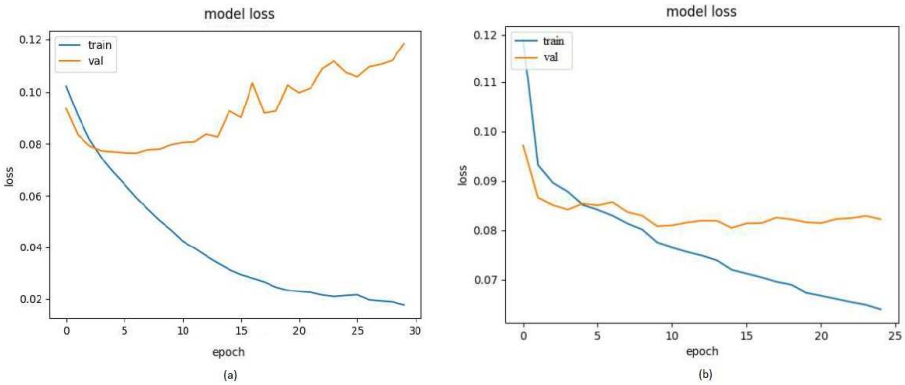



Figure 10. (a) The loss curve of ViraMiner, (b) The loss curve of ResVG

## 5 Conclusion

This paper is proposed to prove the performance of the ResNet model to solve genomic problems and to tackle the overfitting problem presented in the CNN models. Therefore, we proposed two ResNet models

called ResSEN and ResVG. The first model aims to predict the presence of super-enhancers on genome scale, it was tested and evaluated using 11100 samples composed each one of 36 features of mESC datasets taken from Gene Expression Ominibus (GEO). The second model aims to identify viral genomes, it was evaluated using 264049 metagenomic sequences of the size of 300 bp. The obtained results were compared respectively with those of two CNN models called DeepSEN [5] and ViraMiner [59] models. Comparisons showed that the overfitting problem is clearly disappeared in the ResSEN model and improved in the ResVG model. The final results showed also that the ResSEN is better than the DeepSEN for predicting super-enhancers and ResVG is better than the ViraMiner for identifying viral genomes but it can be more enhanced in the future by testing another optimized model of a CNN, like DenseNet or SENet. Finally, we conclude that the ResNet model can be a best solution for some genomic problems and it deserves to be tested in this domain.

## References

[1] M. Alazab et al., "COVID-19 Prediction and Detection Using Deep Learning," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 12, pp. 168–181, 2020.

[2] S. Albaradei et al., "Splice2Deep: An ensemble of deep convolutional neural networks for improved splice site prediction in genomic DNA," *Gene X*, vol. 5, 2020.

[3] B. Alipanahi, A. Delong, M. Weirauch, and B.J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnol*, vol. 33, pp. 831-–838, 2015.

[4] J.E. Bradner, D. Hnisz, and R.A. Young, "Transcriptional addiction in cancer," *Cell*, vol. 168, pp. 629-–643, 2017.

[5] H. Bu, J. Hao, Y. Gan, et al., "DEEPSEN: a convolutional neural network based method for super-enhancer prediction," *BMC Bioinformatics*, vol. 20, 2019.

[6] H. Bu, J. Hao, and Y. Gan, *DEEPSEN code*, 2019. [Online]. Available: https://github.com/1991Troy/DEEPSEN.

[7] D. Chen, F. Hu, G. Nian, and T. Yang, "Deep Residual Learning for Nonlinear Regression," *Entropy*, vol. 22, no. 2, Article ID: 193, 2020. https://doi.org/10.3390/e22020193.

[8] S. Chen, Q. Jia, Q., Y. Tan, Y. Li, and F. Tang, "Oncogenic super-enhancer formation in tumorigenesis and its molecular mechanisms," *Experimental & Molecular Medicine*, vol. 52, pp. 713-–723, 2020.

[9] T. Ching et al., "Opportunities and obstacles for deep learning in biology and medicine," *Journal of the Royal Society Interface*, vol. 15, no. 141, Article ID: 20170387, 2018. DOI: 10.1098/rsif.2017.0387.

[10] A. Esteva et al., "A guide to deep learning in healthcare," *Nature Medecine*, vol. 25, no.1, 2019.

[11] Y. Furusho and K. Ikeda, "ResNet and Batch-normalization Improve Data Separability," in *Proceedings of Machine Learning Research, vol. 101*, pp. 94–108, 2019.

[12] S.R. Grossman et al., "Identifying recent adaptations in large-scale genomic data," *Cell*, vol. 152, pp. 703-–713, 2013.

[13] Y. He, W. Long, and Q. Liu, "Targeting Super-Enhancers as a Therapeutic Strategy for Cancer Treatment," *Frontiers in Pharmacology*, vol. 10, 2019.

[14] D. Hnisz, B.J. Abraham et al., "Super-enhancers in the control of cell identity and disease," *Cell*, vol. 155, no. 4, pp. 934–947, 2013.

[15] A. Holzinger and I. Jurisica, "Knowledge Discovery and Data Mining in Biomedical Informatics: The Future Is in Integrative, Interactive Machine Learning Solutions," in *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics* (Lecture Notes in Computer Science, vol. 8401), A. Holzinger and I. Jurisica, Eds. 2014.

[16] J. Huang et al., "Dissecting super-enhancer hierarchy based on chromatin interactions," *Nature Communications*, vol. 9, no. 943, 2018.

[17] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv*, 2015. [Online]. Available: arXiv:1502.03167.

[18] Y. Jia, W. Chng, and J. Zhou, "Super-enhancers: critical roles and therapeutic targets in hematologic malignancies," *Journal of Hematology and Oncology*, vol.12, 2019.

[19] A. Khan and X. Zhang, "Integrative modeling reveals key chromatin and sequence signatures predicting super-enhancers," Scientific Reports, vol. 9, pp. 1–15, 2019. DOI: 10.1038/s41598-019-38979-9.

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2014.

[21] T.I. Lee and R.A. Young, "Transcriptional regulation and its misregulation in disease," *Cell*, vol. 152, pp. 1237–1251, 2013.

[22] G. Litjens el al., "A survey on deep learning in medical image analysis," *Medical Image Analysis*, vol. 42, pp. 60–88, 2017.

[23] J. Lu et al., "MICAL2 mediates p53 ubiquitin degradation through oxidating p53 methionine 40 and 160 and promotes colorectal cancer malignance," *Theranostics*, vol. 8, no. 19, pp. 5289–5306, 2018.

[24] P. Mamoshina, A. Vieira, E. Putin, and A. Zhavoronkov, "Applications of deep learning in biomedicine," *Molecular Pharmaceutics*, vol. 13, no. 5, pp. 1445–1454, 2016.

[25] M. R. Mansour et al., "Oncogene regulation. An oncogenic super-enhancer formed through somatic mutation of a noncoding intergenic element," *Science* (New York, N.Y.), vol. 346, no. 6215, pp. 1373–1377, 2014.

[26] M.F et al., "Super-enhancers maintain renin-expressing cell identity and memory to preserve multi-system homeostasis," *Journal Clinical Investigation*, vol. 128, no. 11, pp. 4787–4803, 2018.

[27] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Brief Bioinform*, vol. 18, no. 5, pp. 851–869, 2017.

[28] H.H. Ng and M.A. Surani, "The transcriptional and signalling networks of pluripotency," *Nature cell biology*, vol. 13, pp. 490–496, 2011.

[29] S.H. Orkin and K. Hochedlinger, "Chromatin connections to pluripotency and cellular reprogramming," *Cell*, vol. 145, pp. 835–850, 2011.

[30] J. Qu et al., "Functions and Clinical Significance of Super-Enhancers in Bone-Related Diseases," *Frontiers in Cell and Developmental Biology*, vol. 8, 2020.

[31] S. Sengupta and R.E. George, "Super-enhancer-driven transcriptional dependencies in cancer," *Trends Cancer*, vol. 3, pp. 269–281, 2017.

[32] F. Tang, Z. Yang, Y. Tan, and Y. Li, "Super-enhancer function and its application in cancer targeted therapy," *NPJ Precision Oncology*, vol. 4, no. 2, 2020.

[33] H. Wang and H. Zheng, "Model Validation, Machine Learning," in *Encyclopedia of Systems Biology*, W. Dubitzky, O. Wolkenhauer, KH. Cho, H. Yokota, Eds. New York, NY: Springer, 2013. https://doi.org/10.1007/978-1-4419-9863-7_233.

[34] R. Wang, Z. Wang, J. Wang, and S. Li, "SpliceFinder: ab initio prediction of splice sites using convolutional neural network," *BMC Bioinformatics*, vol. 20, 2019.

[35] W. Whyte et al., "Master transcription factors and mediator establish super-enhancers at key cell identity genes," *Cell*, vol. 153, no. 2, pp. 307–319, 2013.

[36] W. Xi, J.C. Murray, and Y. Jian, "Super-enhancers in transcriptional regulation and genome organization," *Nucleic Acids Research*, vol.47, no. 22, pp. 11481–11496, 2019.

[37] M. Xu, C. Ning, C. Ting, and J. Rui, "DeepEnhancer: Predicting enhancers by convolutional neural networks," in *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, (Shenzhen), pp. 637–644, 2016.

[38] Y. Xu and R. Goodacre, "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning," *Journal of Analysis and Testing*, vol. 2, no. 3, pp. 249–262, 2018.

[39] R.A. Young, "Control of the embryonic stem cell state," *Cell*, vol. 144, pp. 940–954, 2011.

[40] J. Zhou and O. Troyanskay, "Predicting effects of noncoding variants with deep learning–based sequence model," *Nature Methods*, vol. 12, pp. 931–934, 2015.

[41] H. Zilonga, T. Jinshanabc, W. Zimingb, Z. Kaiac, Z. Linga, and S. Qingling, "Deep learning for image-based cancer detection and diagnosis – A survey," *Pattern Recognition*, vol. 83, pp. 134–149, 2018.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv*, 2015. [Online]. Available: arXiv:1512.03385.

[43] L. Wang, X. Zhong, S. Wang, S. et al., "A novel end-to-end method to predict RNA secondary structure profile based on bidirectional LSTM and residual neural network," *BMC Bioinformatics*, vol. 22, 2021.

[44] Y. Li., J. Hu, C. Zhang, D.J. Yu, and Y. Zhang, "ResPRE: high-accuracy protein contact prediction by coupling precision matrix with deep residual neural networks," *Bioinformatics*, vol. 35, no. 22, pp. 4647–4655, Nov. 2019.

[45] S. Sun, W. Wang, Z. Peng, and J. Yang, "RNA inter-nucleotide 3D closeness prediction by deep residual neural networks," *Bioinformatics*, vol. 37, no. 8, pp. 1093–1098, Apr, 2021.

[46] M.H. Shuvo, S. Bhattacharya, and D. Bhattacharya, "QDeep: distance-based protein model quality estimation by residue-level ensemble error classifications using stacked deep residual neural networks," *Bioinformatics*, vol 36, pp. i285–i291, July, 2020.

[47] Y. Zhang, X. Zhou, and X. Cai, "Predicting Gene Expression from DNA Sequence using Residual Neural Network," *bioRxiv*, 2020. DOI: https://doi.org/10.1101/2020.06.21.163956.

[48] J. Li and J. Xu, "Study of real-valued distance prediction for protein structure prediction with deep learning," *Bioinformatics*, vol. 37, no. 19, pp. 3197–3203, Oct. 2021.

[49] H. Zhang and Y. Shen, "Template-based prediction of protein structure with deep learning," *BMC Genomics*, vol. 21, Supplement 11, Article number: 878, Dec. 2020. https://doi.org/10.1186/s12864-020-07249-8.

[50] J. Kandel, H. Tayara, and K.T. Chong, "PUResNet: prediction of protein-ligand binding sites using deep residual neural network," *J Cheminform*, vol. 13, no. 65, 2021.

[51] S. Sharma, A. Juneja, and N. Sharma, "Using Deep Convolutional Neural Network in Computer Vision for Real-World Scene Classification," in *IEEE 8th International Advance Computing Conference*, pp. 284–289, 2018.

[52] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 7068349, 13 pages, 2018. https://doi.org/10.1155/2018/7068349.

[53] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Feb. 2021.

[54] K.M. Tarwani and S. Edem, "Survey on recurrent neural network in natural language processing," *Int. J. Eng. Trends Technol*, vol. 48, pp. 301–304, 2017.

[55] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of CNN and RNN for natural language processing," *arXiv* preprint, arXiv:1702.01923, 2017.

[56] H. Purwins et al., "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[57] C. Tian, Y. Xu, L. Fei, J. Wang, J. Wen, and N. Luo, "Enhanced CNN for image denoising," *CAAI Transactions on Intelligence Technology*, vol. 4, no. 1, pp. 17–23, 2019.

[58] L. Wen, Y. Dong, and L. Gao, "A new ensemble residual convolutional neural network for remaining useful life estimation," *Math. Biosci. Eng*, vol. 16, no. 2, pp. 862–880, 2019.

[59] A. Tampuu, Z. Bzhalava Z., J. Dillner, and R. Vicente, "ViraMiner: Deep learning on raw DNA sequences for identifying viral genomes in human samples," *PLoS ONE*, vol. 14, no. 9. 2019.

[60] S. Dávila-Ramos Sonia et al., "A Review on Viral Metagenomics in Extreme Environments," *Frontiers in Microbiology*, vol. 10, 2019.

[61] E.L. Delwart, "Viral metagenomics," *Rev Med Virol.*, vol. 17, no. 2, pp. 115–131, 2007.

[62] T.M. Santiago-Rodriguez and E.B. Hollister, "Potential Applications of Human Viral Metagenomics and Reference Materials: Considerations for Current and Future Viruses," *Appl Environ Microbiol*, vol. 86, no. 22, e01794-20, Oct, 2020.

[63] P. Bernardo, E. Albina, M. Eloit, and P. Roumagnac, "Pathology and viral metagenomics, a recent history," *Medecine sciences*, vol. 29, no. 5, pp. 501–508, 2013.

[64] J. Ren, N.A. Ahlgren, Y.Y. Lu, J.A. Fuhrman, and F. Sun, "VirFinder: a novel k-mer based tool for identifying viral sequences from assembled metagenomic data," *Microbiome*, vol. 5, no. 69, 2017.

[65] J. Ren et al., "Identifying viruses from metagenomic data by deep learning," arXiv:1806.07810, 2020.

[66] A. Tampuu, Z. Bzhalava, J. Dillner, and R. Vicente, *ViraMiner*, 2019. https://github.com/NeuroCSUT/ViraMiner.

Sara Sabba, Meroua Smara,
Mehdi Benhacine, Loubna Terra,
Zine Eddine Terra

Received November 28, 2021
Revised January 04, 2022
Accepted February 19, 2022

Sara Sabba
Department of Software Technologies and Information Systems,
Laboratory of Data Science and Artificial Intelligence(LISIA),
Abdelhamid Mahri University, Constantine 2, Algeria.
E–mail: sara.sabba@univ-constantine2.dz

Meroua Smara, Mehdi Benhacine, Loubna Terra, Zine Eddine Terra
Faculty of New Technologies of Information and Communication
Department of Software Technologies and Information Systems,
Abdelhamid Mahri University, Constantine 2, Algeria.
E–mail: meroua.smara@univ-constantine2.dz
E–mail: mehdi.benhacine@univ-constantine2.dz
E–mail: loubna.terra@univ-constantine2.dz
E–mail: zineeddine.terra@univ-constantine2.dz