Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05673-1

S.I. : HEALTHCARE ANALYTICS

CircNet: an encoder–decoder-based convolution neural network (CNN) for circular RNA identification

Marco Stricker[1,2] · Muhammad Nabeel Asim[1,2] · Andreas Dengel[1,2] · Sheraz Ahmed[1]

Received: 21 June 2020 / Accepted: 28 December 2020

Abstract
Discrimination of circular RNA from long non-coding RNA is important to understand its role in different biological processes, disease prediction and cure. Identifying circular RNA through manual laboratories work is expensive, time-consuming and prone to errors. Development of computational methodologies for identification of circular RNA is an active area of research. State-of-the-art circular RNA identification methodologies make use of handcrafted features, which not only increase the feature space, but also extract irrelevant and redundant features. The paper in hand proposes an end-to-end deep learning-based framework named as CircNet, which does not require any handcrafted features. It takes raw RNA sequence as an input and utilises encoder–decoder based convolutional operations to learn lower-dimensional latent representation. This latent representation is further passed to another convolutional architecture to extract discriminative features followed by a classification layer. We performed extensive experimentation to highlight different regions of genome sequence that preserve the most important information for identifying circular RNAs. CircNet significantly outperforms state-of-the-art approaches with a considerable margin 10.29% in terms $F1$ measure.

Keywords Circular RNA classification · Machine learning · Deep learning · Autoencoder

1 Introduction

Ribonucleic acid (RNA) is an essential molecule for living entities and is involved in multifarious biological processes, such as translation, sponging, gene regulation and splicing [1, 2]. Four basic nucleotides, namely guanine (G), uracil (U), adenine (A) and cytosine (C) define the basic structure of RNA molecules [3], where structure means to have knowledge about its biological properties. Involvement of RNA molecules in different biological functions and its importance in different diseases attracts many researchers to analyse RNA molecules in more detail to find new functions and roles in biological processes [4–6]. To understand different biological roles of RNA molecules, it is being classified into several categories based on its structure, physical and chemical properties [7]. Generally, RNA molecules are categorised into coding and non-coding RNA classes.

Previously, non-coding RNAs were considered junk of code and thought to not participate in the process of developing proteins [4, 8, 9]. However, lately it was discovered that they not only participate in the development of proteins, but also control the process in which proteins gets produced. Furthermore, their involvement in various biological processes, such as translation, splicing, gene regulation and sponging, was discovered [1, 2]. Research findings about the biological role of non-coding RNAs and their importance for disease predictions attracts researchers to explore this domain more precisely [10].

Furthermore, non-coding RNA can be further categorised into small non-coding RNA and long non-coding RNA based on their sequence length. If the length is shorter than 200 nucleotides, it is considered small, else it is

✉ Muhammad Nabeel Asim
Muhammad_Nabeel.Asim@dfki.de

Marco Stricker
Marco.Stricker@dfki.de

Andreas Dengel
Andreas.Dengel@dfki.de

Sheraz Ahmed
Sheraz.Ahmed@dfki.de

[1] German Research Center for Artificial Intelligence (DFKI), 67663 Kaiserslautern, Germany

[2] TU Kaiserslautern, 67663 Kaiserslautern, Germany

Published online: 21 January 2021

defined as long [11]. Long non-coding RNA is further diversified based on the structure and various physical and chemical properties. In this case, two main categories exist, linear and circular, which in turn have multiple subcategories. On the other hand, small non-coding RNA is categorised into three main classes, namely Cis-regulatory, Gene and Intron. These three classes are further categorised into several subtypes. A graphical representation of non-coding RNA hierarchy is shown in Fig. 1.

Following recent research about non-coding RNAs and findings about their role in biological processes, disease prediction and use in therapies, among different types of non-coding RNA, circular RNA is a more attractive research area [12]. Moreover, the identification of drugs targeting the regulatory circuits of functional RNAs depends on knowning its family, a task which is known as RNA sequence classification. In order to perform circular RNA classification, it is necessary to know about the formation of circular RNA. In circular RNA formation, first DNA is transcribed into a precursor messenger RNA (pre-mRNA) [12]. This pre-mRNA consists of introns and exons [13], where the intron sections are removed by a process called splicing, which creates the mature messenger RNA [13]. During the process of pre-RNA splicing, alternative events may occur [14]. This includes back-splicing, a special splicing alternative, which is responsible for creating circular RNA [15]. Compared to linear RNA, the circular structure offers benefits, such as efficient copying, high stability and being able to change the order of genetic information contained in DNA [15]. A slight change in the flow of the formation of circular RNA leads towards failure of various biological processes and development of diseases, such as cancer, Alzheimer or Parkinson [10, 16, 17].

Circular RNA is involved in various biological processes; however, it is still not fully understood. Therefore, precise identification of circular RNA is necessary in order to investigate those processes in detail [7, 9, 18, 19]. Classification can help in the case of diagnosis, if this specific circular RNA is promoting said disease, while it can also help with treatments, by using RNA's gene regulating potential which is able to suppress the disease [17, 20, 21].

One way to classify or identify circular RNA is via biological experiments in a laboratory, as it is done by Zaghlool et al. [22] and Zirkel et al. [23]. Unfortunately, performing such experiments suffers from multiple drawbacks such as relatively low appearance rate of circRNA compared to other RNA and the similarity of sequence information with nonlinear RNA [22]. Following the instructions from Zirkel et al. [23], it is obvious that providing all chemical materials is costly and performing all steps is also time-consuming [23]. Furthermore, performing laboratory experiments is always error prone, as it can be seen in Zaghlool et al. [22] where a low reproducibility rate of different methods is reported.

Thanks to high-throughput technologies which produce large amount of nucleotide sequencing data [12, 24], researchers from bioinformatics domain utilise said data in machine and deep learning approaches. Various computational methods have been proposed for tasks such as circular RNA classification [8, 11, 25], predicting DNA methylation level [26, 27], nucleosome position prediction [28], compound–protein interaction prediction [29], chemical–chemical interaction prediction [30] and DNA histone analysis and prediction [31, 32]. Due to the data availability and the previously mentioned drawbacks of

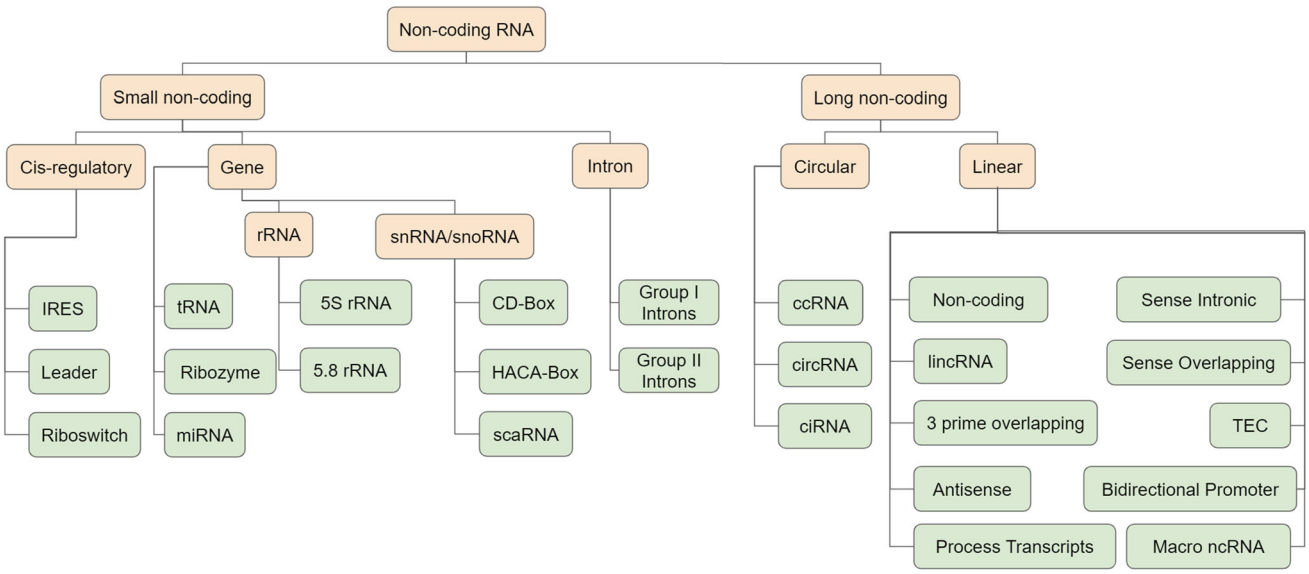


Fig. 1 Non-coding RNA Taxonomy. Figure adapted from [7]

laboratory experiments, the application of machine learning algorithms has become an active area of research. To the best of our knowledge, there are currently three approaches categorising between circular RNA and other long non-coding RNA. The first approach PredcircRNA [8] extracts different features, such as graph features, conservation information, sequence compositions, ALU, tandem repeat, SNP density and ORF features. Based on this extraction a multiple kernel learning method is applied, which works similar to a nonlinear SVM. This classifier learns a linear weight combination of several kernels, where a single kernel transforms the input feature representation into a higher-dimensional space. Data which is not linearly separable in its original space will become linearly separable in higher-dimensional space. After this transformation SVM is applied for classification.

As the previous method H-ELM [25] also extracts similar features. In total 188 features are extracted from the sequence. However, as an additional step, the authors use the minimum redundancy maximum relevance (mRMR) method in order to analyse them. Following this, a combination of the incremental feature selection method and hierarchical extreme learning is applied to find a set of discriminative features. The main disadvantages of both methods are the inability to capture the structure of circular RNA and properly utilise the co-occurrence of trinucleotides [11]. In our opinion, another drawback is the use of so many handcrafted features, which make learning more complex and difficult.

On the other hand circDeep [11] greatly reduces the number of used features to three. It fuses a conservation score, an RCM (Reverse Complement Matching) score of the sequences adjacent to the circular RNA and a feature, which is extracted from a combination of an ACNN (Asymmetric Convolution Neural Network) and BLSTM (Bidirectional Long Short-Term Memory). Using this approach the authors were able to greatly improve the performance. Unfortunately, this approach still has two drawbacks. First, the extraction of RCM features is highly time-consuming and it still utilises two handcrafted features. However, the strength of current deep learning is being able to find the relevant features by itself, which creates an independence from manually handcrafted and selected features.

Recent research about genome sequence analysis has proved that deep learning-based methodologies perform better when they are fed with raw DNA or RNA sequences as compared to their performance when they are fed with manually extracted features. Asima et al. [33] proposed an end-to-end deep learning-based approach for the classification of small non-coding RNA classification. Based on the experimental results they concluded that when deep learning methodologies are fed with handcrafted features, their performance decreased because during the process of feature extraction important information about occurrences and positions of nucleotides may get lost. Their experimental results proved that deep learning methodologies perform better by extracting more discriminative features form raw sequences based on the position and occurrences of basic nucleotides. Moreover, there also exist several other deep learning methodologies which achieve state-of-the-art performance by processing raw RNA sequences such as classification of long non-coding RNA and subcellular localisation of lncRNA and micro-RNA. Moreover, use of autoencoder for better representation learning is another active area of research. There exist several computation methodologies which use encoder decoder-based architecture for the tasks of translation [34] or driver assistance systems [35].

In order to improve the performance of circular RNA identification, we propose a two-stage classification methodology where at first stage we utilise an encoder decoder approach for the extraction of latent space and at second stage, by utilising learned representation, a convolutional neural network is used for the extraction of discriminative features. Discriminative features are fed to a fully connected layer for the discrimination of circular RNA from other long non-coding RNA. Lastly, in order to explore different regions of genome which contain more important information about the identification of circular RNA, we performed extensive experiments with combinations of different sequence lengths, scaling methods and number of added adjacent nucleotides. CircNet achieved 98.28%, 98.62%, 96.35% and 97.75% in terms of accuracy, $F1$, MCC and specificity, respectively, and outperformed previous state-of-the-art computational-based approaches [8, 11, 25].

2 Materials and methods

This section briefly describes proposed CircNet approach along with benchmark dataset and evaluation measures that are used to evaluate the performance.

2.1 Benchmark dataset

In order to evaluate the integrity of proposed CircNet approach we performed experimentation on the publicly available benchmark dataset provided by Chaabane et al. [11]. It consists of two classes, circular RNA and other lncRNA. Circular RNA are defined as the positive samples, while the other lncRNA are defined as negative. CircRNADb was used to extract the 31939 positive samples [36]. On the other hand, the GENCODE dataset was used to extract 19683 negative samples [37]. More details about

the dataset, such as the minimal-, maximal-, average sequence length and the standard deviation of all sequences, can be seen in Table 1.

2.2 Proposed methodology

We propose a two-stage classification methodology, where at the first stage we learn discriminative features by utilising an encoder decoder architecture and at second stage the learned features are passed to a convolutional neural network for the extraction of more discriminative features and to perform classification between circRNA and other long non-coding RNA(lncRNA). The encoder performs convolution and pooling, while the decoder is responsible for deconvolution and un-pooling/up-sampling to reconstruct the original raw sequence. The key idea is to apply the encoder-based convolutional operations to learn sequence representation in less space, while the up-sampling decoder network makes sure whether the sequence can be reconstructed from the learned space. This architecture substantially reduces the number of trainable parameters and reuses the encoder's pooling indices for the discrimination between circular and lncRNA. A brief description of encoder decoder architecture is given in Sect. 1, and deep learning classifier is described in Sect. 2.2.3. The process of forming circRNA includes different intron and exon regions. In order to clearly understand the area of genome that contains more important information about the identification of circRNA we take different segments of the genome which are briefly described below in the preprocessing stage, Sect. 2.2.1.

2.2.1 Preprocessing

The dataset provided by Chaabane et al. [11] contains the positional information of the RNA sequences in the human genome, i.e. the start and end location of the long non-coding RNA. In order to extract the nucleotide sequence, the location information can be used on a dataset provided by the UCSC Genome Browser [38]. Besides extracting the sequence defined by the start and end location, we extend the sequence by also including a certain amount of adjacent nucleotides, since those regions might also embed valuable features regarding circular RNA classification [39, 40]. We define adjacent nucleotides as nucleotides which appear directly after or before the previously mentioned start and end location. The concept of adjacent nucleotides is illustrated in Fig. 2.

As it can be seen in Table 1, the sequences of circular RNA dataset vary greatly in length from short to very long sequences. Furthermore, deep learning models based on convolutional neural networks require all input samples to have same length of sequence. Here we set the length of sequence equal to M and apply zero padding for the sequences which are smaller than the defined sequence length and truncate the nucleotides from sequences longer than the fixed length. We performed three experiments of changing the size of the sequences to a predefined length M. We apply three padding approaches, denoted as *post*, *pre* and *middle*. In *post*-padding we remove all nucleotides appearing after the *Mth* nucleotide. If the sequence is shorter than M nucleotides, an additional *zero Z* symbol is added at the end of the sequence as many times as is needed to achieve the predefined length. On the other hand *pre*-padding removes or adds nucleotides at the beginning of the sequence in order to scale the length to size M. In our last approach, denoted as *middle* the first and last $M/2$ nucleotides of a sequence are kept, while removing or adding nucleotides in between. An illustration of said approaches can be seen in Table 2.

As deep learning methodologies require data in real number format, we transform each sequence in one-hot encoded representation which is used as the input for our framework. Furthermore, we extract the sequence from the genome dataset based on positional information, which also includes the letter N in addition to the four nucleotides A, C, G and U. Note that due to ambiguity between nucleotides, an exact identification is not always possible. Therefore, the additional symbol N represents either A, U, C or G. Because we are interested in the positions of nucleotides rather than removing it, we give it a one-hot vector representation. In one-hot encoding every nucleotide is represented by a vector of five bits, where four bits are 0 and one bit is 1. The position of the 1 bit is always the same for a specific nucleotide. Using this methodology adenine is represented as $A = [1, 0, 0, 0, 0]$, Cytosine

Table 1 Statistics of benchmark dataset, where minimal sequence length represents the length of shortest sequence and maximal sequence length denotes the longest sequence

Measure	Positive class	Negative class	Both classes
Minimal sequence length	201	204	201
Maximal sequence length	3,050,672	1,536,213	3,050,672
Average sequence length	19,924	18,653	19,439
Standard deviation of sequence lengths	34,439	47,025	39,716

On the other hand average and standard deviation of sequence length calculate the mean and the standard deviation of the sequences in the corresponding classes

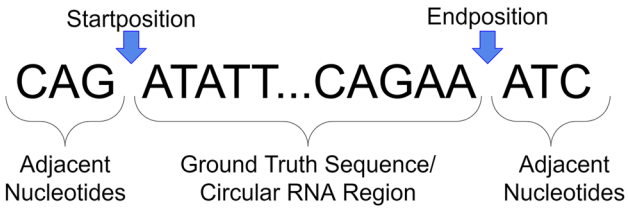


Fig. 2 Considering the circular RNA hsa_circ_00001, defined by its location chr1-230350957-230357321 in bed format, then this figure illustrates the location of three adjacent nucleotides

Table 2 Two different sequences ATAG and ATATGUAT are being scaled to length 6 by either addition or removal with all three different methods

	Pre	Middle	Post
Addition	ZZATAG	ATZZAG	ATAGZZ
Removal	ATGUAT	ATAUAT	ATATGU

$C = [0,0,0,1,0]$, Guanine $G = [0,0,1,0,0]$, Uracil $U = [0,1,0,0,0]$, $N = [0,0,0,0,1]$ and zero symbol $Z = [0,0,0,0,0]$.

2.2.2 Latent space extraction using autoencoder

We utilise raw ncRNA sequences for the extraction of latent space features, where each RNA sequence has four basic nucleotides: adenine (A), cytosine (C), guanine (G) and uracil (U). Furthermore, each nucleotide is encoded using one-hot vector encoding, as described in Sect. 2.2.1. A graphical representation of our autoencoder used for latent space learning is shown in Fig. 3.

We use 1d convolutional layers with 128 filters, kernel size 12 and stride size 1. This layer extracts discriminative features based on the nucleotides' occurrences and positions. Further to decrease the dimensionality we employ a maxpooling layer with kernel size 2. Another set of convolutional layer, with 128 filters, kernel size 6 and stride size 1, and maxpooling layer, with kernel size 2, is used to extract more discriminative features. Reconstructing the initial sequence based on the latent space verifies the extraction of the most discriminative features. For this purpose we use the same number of layers in reverse order. The output of each convolutional layer is calculated by:

$$c_{xyf} = \sum_{i=1}^{k} \sum_{j=1}^{5} n_{ksf,ij} w_{ksf,ij} + b_{ksf,ij} \tag{1}$$

where f denotes the fth filter, x and y the indices of the output tensor and k and s define the currently observed patch of the input tensor n given as kernel and stride size, respectively. i and j denote indices inside this patch. Furthermore, w and b define the learned weights and biases. Considering an input length of 200, then the input is given as a 200×5 tensor for the first convolutional layer. Its output is defined by a 200×128 tensor which gets reduced to a 100×128 after applying maxpooling. Maxpooling calculates the output as a tensor where each index xyf is calculated as following:

$$m_{xyf} = \max c_{ksf} \tag{2}$$

where f denotes the fth filter, x and y the indices of the output tensor and k and s define the currently observed patch of the input tensor c given as kernel and stride size, respectively. The second convolutional layer does not change the shapes. However, the second maxpooling layer again halves the shape to 50×128, which is our latent feature representation. The decoder has the same shapes in reverse order. All layers are utilising Relu as the activation function defined by:

$$a(y) = max(0, y) \tag{3}$$

where y is the output of a layer. However, in the last reversal operation of the decoder sigmoid is applied, which is defined by:

$$a(y) = \frac{1}{1 + e^{-y}} \tag{4}$$

2.2.3 Convolutional neural network

After learning the latent space features, we apply a simple convolutional-based classifier, as shown in Fig. 4, on the RNA sequences transformed into said latent feature space.

We apply two one-dimensional convolutional layers with kernel size 3, stride size 1 and 64 and 32 filters for the first and second layers, respectively. Following this we have a dropout layer with a probability of 0.5 and maxpooling with kernel size and stride size equal to 2. Finally a flatten layer followed by a dense layer is used. Note that we do not freeze the weights of the trained encoder in our classification model, but instead fine-tuned the weights during the second training stage.

Convolutional and maxpooling layer behave the same as defined in Sect. 2.2.2. Dropout randomly ignores a fixed percentage of neurons during the optimisation step. Similarly to Sect. 2.2.2, all layers are using Relu as the activation function, besides the last dense layer which applies softmax as defined by:

$$a(y)_i = \frac{e^{y_i}}{\sum_{j=1}^{K} e^{y_j}} \text{ for } i = 1, \ldots K \text{ and } y = (y_i, \ldots, y_K) \in \mathbb{R}^K \tag{5}$$

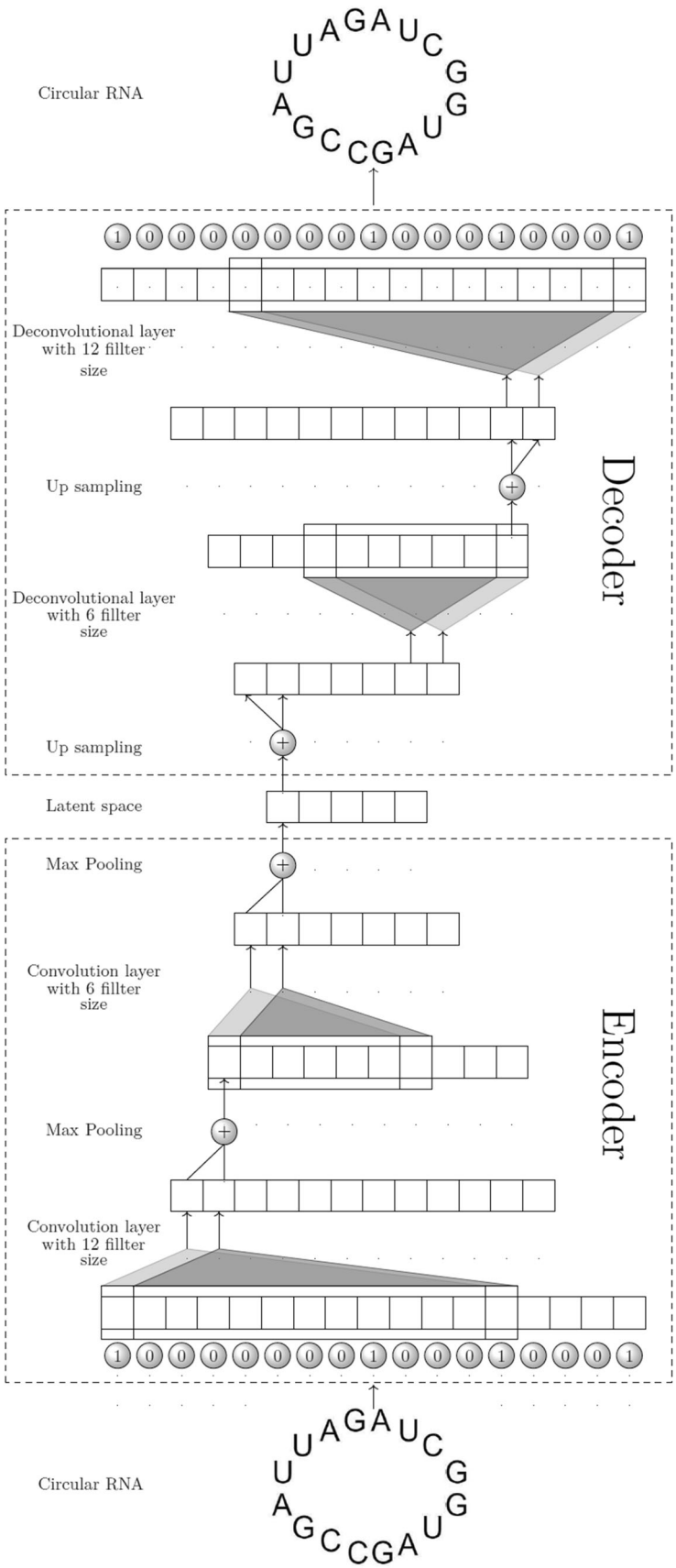


Fig. 3 Graphical representation of the employed autoencoder

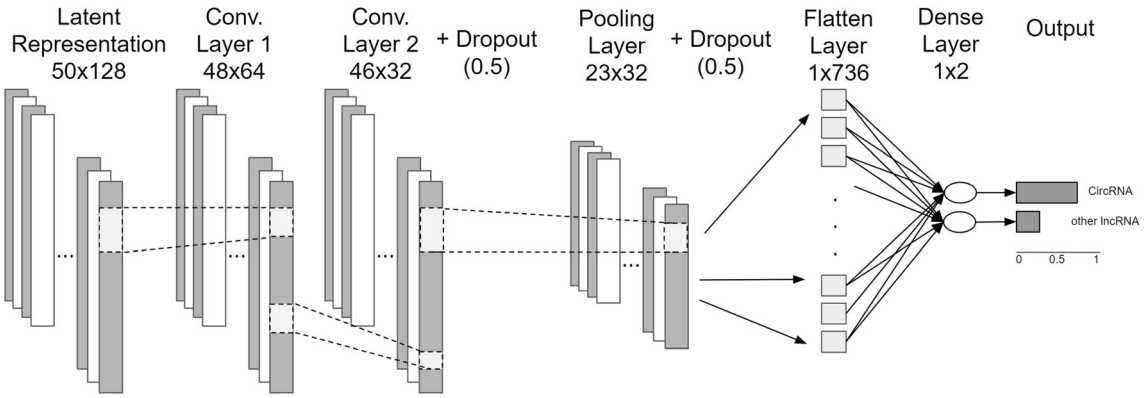


Fig. 4 Graphical representation of the employed classifier

Lets assume we have an input sequence of length 200, our input shape is defined by 200×5 which gets transformed to 50×128 by the encoder, as defined in Sect. 2.2.2. Our convolutional-based classifiers change the shape to 48×64 and 46×32 for the first and second convolutional layers. Dropout does not change our shapes. On the other hand maxpooling reduces the shape to 23×32. Flattening this shape results in a vector of size 736. Lastly the dense layer calculates our final prediction with an output size of 2×1.

2.3 Evaluation measures

To evaluate the performance of CircNet following evaluation criteria, used by previous studies [8, 11, 25] concerning circular RNA classification, are used: accuracy, $f1$-measure, Matthews correlation coefficient, specificity and recall. Furthermore, we also include the ROC curve and AUC as additional evaluation measures. All evaluation measures make use of four basic parameters:

- True Positive (TP): The amount of samples which are correctly classified as positive samples
- True Negative (TN): The amount of samples which are correctly classified as negative samples
- False Positive (FP): The amount of samples which are wrongly classified as positive samples
- False Negative (FN): The amount of samples which are wrongly classified as negative samples

Using these four parameters our evaluation metrics are defined as following:

2.3.1 Accuracy

Accuracy, defined in Eq. 6, is a common metric and describes the ratio of correctly classified samples to the total number classified.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{6}$$

2.3.2 *F*1-measure

$F1$ measure calculates the performance by making use of precision and recall measures. The first one, defined in Eq. 7, describes the ratio of the elements, which are correctly predicted as positive samples compared to all samples that are predicted to be positive. On the other hand recall, defined in Eq. 8, calculates the ratio of elements, which are correctly predicted as positive samples compared to all positive samples in the ground truth data. As it can be seen, these two measures describe different aspects and the $F1$-Measure, defined in Eq. 9, is combining these two measures into one, with equal focus on both.

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{9}$$

2.3.3 Matthews correlation coefficient

Accuracy is considered a good evaluation measure for binary classification [41]. Even though we have a binary classification problem, our dataset is slightly imbalanced with having more positive samples. To precisely evaluate the performance of classification methodology we use MCC measure which is considered to be a better performance evaluator when dataset is imbalanced.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{10}$$

2.3.4 Specificity

Specificity, defined in Eq. 11, describes the ratio of how many negative elements were truly classified as negative.

$$Specificity = \frac{TN}{TN + FP} \tag{11}$$

2.3.5 ROC curve and AUC

ROC (receiver operating characteristic curve) is the most widely used evaluation metric to graphically show the performance of a model. It has been extensively used in many domains like signal detection, machine learning and medical diagnosis [42]. Furthermore, it is also being utilised in sequence classification tasks, similar to our case, such as predicting the protein binding capabilities of circRNA [43, 44], protein–protein interaction prediction [45] and protein virus interaction prediction [46]. The ROC curve plots the true positive rate (TPR, it is also known as recall, as described in Eq. 8) against the false positive rate (FPR), as described in Eq. 12, at all classification thresholds. The classification threshold describes the minimum confidence score, outputted by a classifier in order to categorise a sample as positive. Decreasing the threshold, will improve the TPR and FPR [42], which shows how the model performs under different conditions, differentiating it from other evaluation metrics which mostly only offer a single value to estimate the capabilities.

This graphical representation can be summarised in a single value, namely the AUC (Area Under the ROC Curve), which measures the area underneath the curve. Statistically, it can be interpreted as the expectation that a random positive sample achieves a higher confidence score compared to a random negative sample [42].

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

3 Experimental set-up and results

In order to ensure a fair comparison of CircNet with state-of-the-art circular RNA classification approach, we performed experimentation with standard data splits provided by Chaabane et al. [11] where benchmark dataset has 75% train, 15% test and 10% validation sets.

The learnable parameters of CircNet are optimised with RMSprop, with an initial learning rate of 0.001 and the mean squared error (MSE) as the loss function. RMSprop is defined as following:

$$RMSprop: \tag{13}$$

$$v_t = \rho \cdot v_{t-1} + (1 - \rho) \cdot g_t^2 \tag{14}$$

$$\Delta w_t = -\frac{\alpha}{\sqrt{v_t + \epsilon}} \cdot g_t \tag{15}$$

$$w_{t+1} = w_t + \Delta w_t \tag{16}$$

RMSprop is applied on all weights w_i in the network. However, since the procedure is equal for all w_i, i has been dropped in the equation. w_t denotes a weight at timestep t, α the learning rate, ϵ is a small positive value ensuring that we do not divide by 0, g_t the gradient at timestep t and ρ is a predefined hyperparameter.

The loss function MSE is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \tag{17}$$

where n is the number of samples, y_i the ground truth values and $\hat{y_i}$ the corresponding prediction.

CircNet is trained for 100 epochs with a batch size of 128. In addition, if the validation loss does not improve for 7 epochs, the learning rate is reduced by a factor of 0.1. However, if the loss does not decrease after 10 epochs we stop our training process in a procedure known as early stopping. Lastly, only the network weights corresponding to the best validation loss are saved and used in the second training stage. At the second training stage we employ the same optimiser and use the same parameter settings as described at the first stage. But the second stage learns a classification problem, for which we use binary crossentropy (BCE) as the loss function.

$$BCE = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log(\hat{y_i}) + (1 - y_i) \cdot \log(1 - \hat{y_i}) \tag{18}$$

where n is the number of samples, y_i the ground truth values and $\hat{y_i}$ the corresponding prediction.

Using the above-defined settings we performed different experiments on the dataset. As briefly described in Sect. 2.2.1, where we defined the concept of adjacent nucleotides, we process the input sequences in three different ways. First, we use a different length of adjacent nucleotides, ranging from 0 to 100. Experiments with 0 adjacent nucleotides are performed in order to verify the claim that important features can be extracted from sequence regions of genome adjacent to circular RNA sequence. The different extension lengths are combined with the three different scaling methods *middle, pre and post*, as described in Sect. 2.2.1, which in turn are combined with scaling to different sequence lengths of 200, 500 or 1000. Considering all possible settings, 26 experiments have been performed and the results are summarised in Table 3.

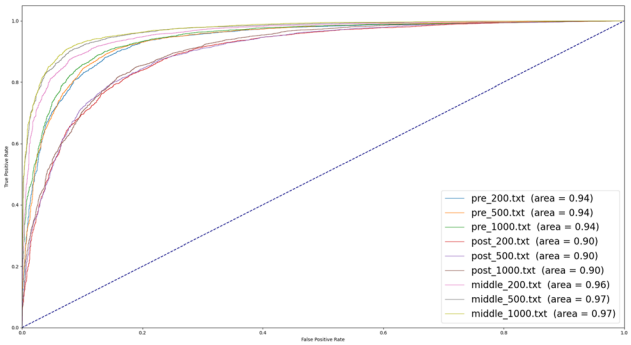
Table 3 Performance evaluation of CircNet based on different number of adjacent nucleotides, different scaling methods and sequence length

Number of adjacent nucleotides	Scaling	Seq. Len.	Acc.	*F*1	MCC	Spec.
0	Middle	200	0.8985	0.9192	0.7832	0.8462
		500	0.9063	0.9266	0.7997	0.8278
		1000	0.9134	0.9315	0.8148	0.8544
	Post	200	0.8301	0.8709	0.6335	0.6774
		500	0.8293	0.8689	0.6315	0.6941
		1000	0.8372	0.8705	0.6518	0.7642
	Pre	200	0.8802	0.9052	0.7434	0.8115
		500	0.8831	0.9067	0.7506	0.8299
		1000	0.8751	0.9040	0.7328	0.7567
50	**Middle**	200	0.9827	0.9860	0.9633	0.9813
		500	0.9818	0.9853	0.9615	**0.9826**
		1000	0.9813	0.9849	0.9602	0.9755
	Post	200	0.9771	0.9816	0.9514	0.9653
		500	0.9771	0.9816	0.9514	0.9650
		1000	0.9768	0.9813	0.9506	0.9646
	Pre	200	0.9700	0.9758	0.9366	0.9670
		500	0.9702	0.9759	0.9369	0.9677
		1000	0.9702	0.9759	0.9368	0.9656
100	**Middle**	**200**	**0.9810**	**0.9847**	**0.9598**	**0.9855**
		500	0.9823	0.9858	0.9624	0.9724
		1000	**0.9828**	**0.9862**	**0.9635**	0.9775
	Post	200	0.9770	0.9815	0.9511	0.9639
		500	0.9773	0.9818	0.9517	0.9636
		1000	0.9775	0.9819	0.9523	0.9677
	Pre	200	0.9703	0.9760	0.9371	0.9660
		500	0.9702	0.9759	0.9368	0.9639
		1000	0.9698	0.9756	0.9360	0.9650

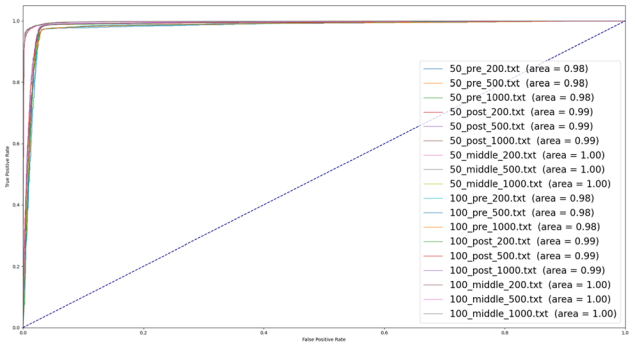
All experiments are evaluated on accuracy, *F*1, MCC and specificity as described in Sect. 2.3. Bolded are the best evaluation scores and common experiment settings responsible for the best scores

From Table 3, it can be concluded that when CircNet is fed with only circular RNA sequences using the three different padding schemes post, pre and middle, the performance of CircNet is better when segments of sequences are extracted from the start and end of the sequence. This also proves that in a sequence more important information lies at the beginning and end of a sequence. Moreover, it can also be summarised that results of the CircNet approach continuously improves with increased sequence length. As when CircNet is fed with an input of 200 nucleotides and padding at the middle, it only achieves 0.9827, 0.9860, 0.9633 and 0.9813 in terms of accuracy, *f*1, mcc and specificity measure, which was the highest among all three padding approaches. However, when input length increases to 500 nucleotides the performance of all three measures improves. The same scenario holds true when the length is increased to 1000 nucleotides. On the other hand the experimental results also validate that by using adjacent nucleotides performance gets improved. Along with the addition of adjacent nucleotides, here once again middle padding approach performed better as compared to other pre- and post-padding approaches. Comparing the best performing model which does not use adjacent nucleotides with the worst performing model which uses adjacent nucleotides, it shows an increase of 5.64% for accuracy, 4.41% for *F*1, 12.12% for MCC and 10.92% for specificity for the latter model. Furthermore, the *middle* scaling method has also performed superior to the other methods, showing that the most important information is contained at the border sections. In three cases, 100 adjacent nucleotides with 1000 sequence length has shown the best results, while 50 adjacent nucleotides with a sequence length of 500 achieved the best specificity.

Furthermore, we evaluate the integrity of proposed CircNet methodology using ROC curves and the respective AUC values for all performed experiments, which are shown in Fig. 5. In detail, Fig. 5a illustrates the curves when the model was fed with different subparts of original



(a) CircNet performance by feeding different subsequences of the original sequence

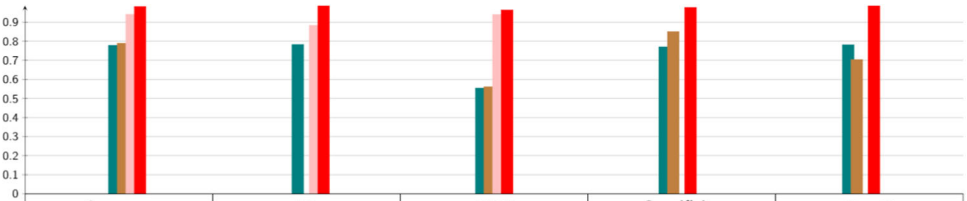


(b) CircNet performance by feeding different subsequences taken from the original sequence and fusing it with genomic subsequences adjacent to circular RNA positions

Fig. 5 CircNet performance in terms of AUROC for different experimental settings by taking subsequences from different positions along with fusion of genome adjacent nucleotide information

sequence and the curves of Fig. 5b represent different subsequences including their extension by adjacent nucleotides. A detailed description of how we define adjacent nucleotides and what our motivation is in using them is given in Sect. 2.2.1. Briefly said, these are nucleotides appearing in front and after the circular RNA sequence in the original genome. Since CircNet requires a fix input size, we experimented with three different methods, which are explained in Sect. 2.2.1, to transform sequences of differing length to the desired number of nucleotides. From Fig. 5 it can be concluded that the approach denoted with middle, in which we extract nucleotides from the beginning and end of the sequence, is superior compared to the circNet performance when it was fed with pre- and post-sequence length selection method. This highlights the fact that the first and last part contains not only enough, but also the most crucial information in order to discriminate circular RNA from non-circular. In the case of not using adjacent nucleotides, middle length scaling approach achieves an AUC of 0.96 in the experimental set-up resulting in its worst AUC, where pre- and post-length selection method have achieved AUC of 0.94 and 0.90, respectively, in their best performing experiment. Moreover, as written in our motivation, important information is contained in adjacent nucleotides, as it can be seen from the AUROC values, where the AUC value is always higher when including adjacent nucleotides compared to using only the original ones. The worst AUC measure in the adjacent case is 0.98, while the best AUC value for the non-adjacent case is 0.97. Visually, this improved performance can also be observed in the ROC curves, since the curves in the adjacent case converge faster to a high true positive rate, compared to the non-adjacent ones. Lastly, the curves representing the adjacent case all behave very similar and are quite close to each other, unlike the non-adjacent case, where many curves vary largely.

Figure 6 illustrates the comparison of our best performing CircNet approach with three previous machine and deep learning-based approaches for the task of circular RNA classification. PredcircRNA [8] approach makes use of too many handcrafted features which number 188. Applying a multiple kernel learning method it only manages produce 77% accuracy and 78.1% $F1$ measure and much less MCC value of only 55.4%. On the other hand H-ELM [25] approach uses the same amount of handcrafted features, but they only selected a subset of those features as an input for a hierarchical extreme learning algorithm. This slightly improves the performance of accuracy and MCC values by around 1%. This approach also improves about 8% in specificity, but does not manage to achieve recall values better than PredcircRNA [8]. CircDeep [11] approach makes use of a combination of only two handcrafted features with a feature representation learned by a deep learning model, based on convolutional neural networks and long short-term memories. Using feature fusion learning methods, this improves the performance values of accuracy, $F1$, MCC with a significant margin of 16.37%, 10.08% and 38.62%, respectively, compared to PredcircRNA [8]. The proposed CircNet approach, which does not use any handcrafted features, significantly improves the performance values of accuracy, $F1$, MCC with a significant margin of 4.11%, 10.29% and 2.33%. We do not compare specificity and recall values of proposed CircNet approach with the circDeep [11] approach as authors do not report the values of these measures. In comparison with other two approaches, namely PredcircRNA [8] and H-ELM [25], the proposed CircNet approach achieves 12.75% and 20.75% improvement in terms of specificity. Similarly, CircNet approach



	Accuracy	F1	MCC	Specificity	Recall
PredicRNA[23]	0.7780	0.7825	0.5540	0.770	0.7810
H-ELM[6]	0.7890	–	0.5610	0.850	0.703
CircDeep[5]	0.9417	0.8833	0.9402	–	–
CircNet	0.9828	0.9861	0.9635	0.9775	0.9860

Fig. 6 Comparison of our best performing methodology with previous circular RNA classification methods based on accuracy, *F*1, MCC, Specificity and Recall. Highlighted are the best scores and the best performing method name

also outperforms both methodologies in terms of recall with a significant margin of 20.5% from PredcircRNA [8] and 28.3% from H-ELM [25] approach.

4 Conclusion

In this work we present a novel deep learning framework, named CircNet, based on a combination of autoencoder and convolutional-based classification for categorising circular RNA from other lncRNA. Unlike previous works, CircNet does not rely on any handcrafted features and instead solely utilises deep learning methodologies. This approach ensures, that the feature representation is more fitting compared to handcrafted features, which often extract redundant and irrelevant features. This is highlighted by the fact, that CircNet outperformed previous state-of-the-art methodology CircDeep [11] by a significant improvement of 4.11%, 10.29% and 2.33% in terms of accuracy, *F*1 and MCC measure. To achieve this, CircNet utilises a two-stage training procedure, where at first stage an encoder–decoder-based architecture learns an accurate lower-dimensional feature representation to describe the long non-coding RNA sequences. Then, at second stage a convolutional neural network classifies the sequences based on their feature representation as calculated by the encoder. Furthermore, we performed extensive experimentation in order to find the most discriminative regions of an RNA sequence by testing different sequence lengths, scaling methods and extension of the sequence. We found that our performance evaluation measures slightly increased with longer sequences. However, the biggest impact on performance is due to the scaling method and extending the sequence. Extension of the sequence is done by also incorporating the adjacent nucleotides, also known as flanking introns, into the RNA sequence used as the input. Lastly, the scaling method is responsible for scaling all sequences to the same length. Our best performing approach, denoted as *middle*, removes or adds nucleotides in the middle of the sequence and not at the beginning or end. Since this approach keeps the beginning and the end of a sequence intact, it shows that the most crucial information in discriminating between circular RNA and other long non-coding RNA is located at them.

Finally, we hope that with an increased performance of distinguishing between circular RNA and other long non-coding RNA, we can help with further understanding the role of circular RNA in biological processes, which in turn help with the diagnosis and treatment of many severe diseases, such as cancer, in a fast, cheap and accurate way compared to laboratory experiments.

Funding Sartorius Artificial Intelligence Lab.

Compliance with ethical standards

Conflicts of interest Corresponding author, on the behalf of all authors declares that no conflict of interest is present.

References

1. Mattick JS, Makunin IV (2006) Non-coding RNA. Hum Mol Genet 15(suppl-1):R17–R29
2. Holdt LM, Kohlmaier A, Teupser D (2018) Molecular roles and function of circular RNAs in eukaryotic cells. Cell Mol Life Sci 75(6):1071–1098
3. Rossi E, Monti F, Bronstein M, Liò P (2019) ncRNA classification with graph convolutional networks. arXiv preprint arXiv:1905.06515
4. Yao D, Zhang L, Zheng M, Sun X, Yan L, Liu P (2018) Circ2Disease: a manually curated database of experimentally validated circRNAs in human disease. Sci Rep 8(1):1–6
5. Razzak MI, Imran M, Xu G (2020) Big data analytics for preventive medicine. Neural Comput Appl 32(9):4417–4451
6. Rehman A, Naz S, Razzak I (2020) Leveraging big data analytics in healthcare enhancement: trends, challenges and opportunities. arXiv preprint arXiv:2004.09010
7. Amin N, McGrath A, Chen Y-PP (2019) Evaluation of deep learning in non-coding RNA classification. Nat Mach Intell 1(5):246–256

8. Pan X, Xiong K (2015) PredcircRNA: computational classification of circular RNA from other long non-coding RNA using hybrid features. Mol BioSyst 11(8):2219–2226
9. Wang Z, Lei X, Fang-Xiang W (2019) Identifying cancer-specific circRNA-RBP binding sites based on deep learning. Molecules 24(22):4035
10. Lee ECS, Elhassan SAM, Lim GPL, Kok WH, Tan SW, Leong EN, Tan SH, Chan EWL, Bhattamisra SK, Rajendran R et al (2019) The roles of circular RNAs in human development and diseases. Biomed Pharmacother 111:198–208
11. Chaabane M, Williams RM, Stephens AT, Park JW (2020) circdeep: deep learning approach for circular RNA classification from other long non-coding RNA. Bioinformatics 36(1):73–80
12. Huang S, Yang B, Chen BJ, Bliim N, Ueberham U, Arendt T, Janitz M (2017) The emerging role of circular RNAs in transcriptome regulation. Genomics 109(5–6):401–407
13. Tilgner H, Knowles DG, Johnson R, Davis CA, Chakrabortty S, Djebali S, Curado J, Snyder M, Gingeras TR, Guigó R (2012) Deep sequencing of subcellular RNA fractions shows splicing to be predominantly co-transcriptional in the human genome but inefficient for lncRNAs. Genome Res 22(9):1616–1625
14. Pan Q, Shai O, Lee LJ, Frey BJ, Blencowe BJ (2008) Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. Nat Genet 40(12):1413
15. Lasda E, Parker R (2014) Circular RNAs: diversity of form and function. RNA 20(12):1829–1842
16. Zhang Z, Yang T, Xiao J (2018) Circular RNAs: promising biomarkers for human diseases. EBioMedicine 34:267–274
17. Bachmayr-Heyda A, Reiner AT, Auer K, Sukhbaatar N, Aust S, Bachleitner-Hofmann T, Mesteri I, Grunt TW, Zeillinger R, Pils D (2015) Correlation of circular RNA abundance with proliferation—exemplified with colorectal and ovarian cancer, idiopathic lung fibrosis and normal human tissues. Sci Rep 5(1):1–10
18. Fiannaca A, La Rosa M, La Paglia L, Rizzo R, Urso A (2017) nRC: non-coding RNA classifier based on structural features. BioData Min 10(1):27
19. Zhang X, Wang J, Li J, Chen W, Liu C (2018) CRlncRC: a machine learning-based method for cancer-related long noncoding RNA identification using integrated features. BMC Med Genomics 11(6):99–112
20. Holdt LM, Kohlmaier A, Teupser D (2018) Circular RNAs as therapeutic agents and targets. Front Physiol 9:1262
21. Li P, Chen S, Chen H, Mo X, Li T, Shao Y, Xiao B, Guo J (2015) Using circular RNA as a novel type of biomarker in the screening of gastric cancer. Clin Chim Acta 444:132–136
22. Zaghlool A, Ameur A, Wu C, Westholm JO, Niazi A, Manivannan M, Bramlett K, Nilsson M, Feuk L (2018) Expression profiling and in situ screening of circular RNAs in human tissues. Sci Rep 8(1):1–12
23. Zirkel A, Papantonis A (2018) Detecting circular RNAs by RNA fluorescence in situ hybridization. In: Circular RNAs. Springer, pp 69–75
24. Xia S, Feng J, Lei L, Jun H, Xia L, Jun Wang Yu, Xiang LL, Zhong S, Han L et al (2017) Comprehensive characterization of tissue-specific circular RNAs in the human and mouse genomes. Briefings Bioinform 18(6):984–992
25. Chen L, Zhang Y-H, Huang G, Pan X, Wang SP, Huang T, Cai Y-D (2018) Discriminating cirRNAs from other lncRNAs using a hierarchical extreme learning machine (H-ELM) algorithm with feature selection. Mol Genet Genomics 293(1):137–149
26. Angermueller C, Lee HJ, Reik W, Stegle O (2017) DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. Genome Biol 18(1):67
27. Wang Y, Liu T, Dong X, Shi H, Zhang C, Mo Y-Y, Wang Z (2016) Predicting DNA methylation state of CPG dinucleotide using genome topological features and deep networks. Sci Rep 6:19598
28. Di Gangi M, Bosco GL, Rizzo R (2018) Deep learning architectures for prediction of nucleosome positioning from sequences data. BMC Bioinform 19(14):418
29. Tian K, Shao M, Wang Y, Guan J, Zhou S (2016) Boosting compound-protein interaction prediction by deep learning. Methods 110:64–72
30. Kwon S, Yoon S (2017) Deepcci: end-to-end deep learning for chemical–chemical interaction prediction. In: Proceedings of the 8th ACM international conference on bioinformatics, computational biology, and health informatics, pp 203–212
31. Singh R, Lanchantin J, Robins G, Qi Y (2016) Deepchrome: deep-learning for predicting gene expression from histone modifications. Bioinformatics 32(17):i639–i648
32. Zhou J, Troyanskaya OG (2015) Predicting effects of noncoding variants with deep learning-based sequence model. Nat Methods 12(10):931–934
33. Asima MN, Malik MI, Dengela A, Ahmed S (2019) A robust and precise convnet for small non-coding RNA classification (RPC-SNRC). arXiv preprint arXiv:1912.11356
34. Cho K, Van Merriënboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder-decoder approaches. arXiv preprint arXiv:1409.1259
35. Yasrab R, Naijie G, Zhang X (2017) An encoder-decoder based convolution neural network (CNN) for future advanced driver assistance system (ADAS). Appl Sci 7(4):312
36. Chen X, Han P, Zhou T, Guo X, Song X, Li Y (2016) circRNADb: a comprehensive database for human circular RNAs with protein-coding annotations. Sci Rep 6(1):1–6
37. Frankish A, Diekhans M, Ferreira A-M, Johnson R, Jungreis I, Loveland J, Mudge JM, Sisu C, Wright J, Armstrong J et al (2019) Gencode reference annotation for the human and mouse genomes. Nucleic Acids Res 47(D1):D766–D773
38. Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D (2002) The human genome browser at UCSC. Genome Res 12(6):996–1006
39. Ivanov A, Memczak S, Wyler E, Torti F, Porath HT, Orejuela MR, Piechotta M, Levanon EY, Landthaler M, Dieterich C et al (2015) Analysis of intron sequences reveals hallmarks of circular RNA biogenesis in animals. Cell Rep 10(2):170–177
40. Wang J, Wang L (2019) Deep learning of the back-splicing code for circular RNA formation. Bioinformatics 35(24):5235–5242
41. Straube S, Krell MM (2014) How to evaluate an agent's behavior to infrequent events? Reliable performance estimation insensitive to class distribution. Front Comput Neurosci 8:43
42. Brzezinski D, Stefanowski J (2017) Prequential AUC: properties of the area under the ROC curve for data streams with concept drift. Knowl Inf Syst 52(2):531–562
43. Zhang K, Pan X, Yang Y, Shen H-B (2019) CRIP: predicting circRNA-RBP-binding sites using a codon-based encoding and hybrid deep neural networks. RNA 25(12):1604–1615
44. Jia C, Yue B, Chen J, Leier A, Li F, Song J (2020) PASSION: an ensemble neural network approach for identifying the binding sites of RBPs on circRNAs. Bioinformatics 36(15):4276–4282. https://doi.org/10.1093/bioinformatics/btaa522
45. Javad Z, Omid Y, Morteza M-N, Reza E, Ali M-N (2013) PPievo: protein–protein interaction prediction from PSSM based evolutionary information. Genomics 102(4):237–242
46. Halder AK, Dutta P, Kundu M, Basu S, Nasipuri M (2018) Review of computational methods for virus–host protein interaction prediction: a case study on novel ebola–human interactions. Briefings Funct Genomics 17(6):381–391