Toward Deep Learning Approaches for Learning Structure Motifs and Classifying Biological Sequences from RNA A-to-I Editing Events

Thi Kieu Khanh Ho, Jeonghwan Gwak

1School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, Republic of Korea
2Department of Software, Korea National University of Transportation, Chungju 27469, Republic of Korea

Corresponding author: Jeonghwan Gwak (james.han.gwak@gmail.com).

This work was supported by Korea National University of Transportation in 2019, the Brain Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2019M3C7A1020406, NRF-2016M3C7A1905477), and the Basic Science Research Program through the NRF funded by the Ministry of Education (NRF-2017R1D1A1B03036423).

ABSTRACT RNA editing is a post-transcriptional RNA sequence modification that alters the mature RNA sequence from its template DNA sequences. RNA editing events are critical in various biological and biochemical mechanisms, and can expand the transcriptomic and proteomic diversity from altered gene regulation to mutations. A-to-I RNA editing is now being vastly detected and quantified on a global scale and gained much attention. A deeper understanding of this process with insufficient genomic annotations and prior knowledge-based filtering steps, such as high-throughput next-generation sequencing techniques, are needed, in addition to data regarding whether an editing location is located in one of the following three main classes of RNA editing sites: ALU in Alu repetitive elements, REP in non-Alu repetitive elements, and NONREP in non-repetitive regions. This study proposes deep learning approaches to ameliorate these issues by learning motif patterns and identifying regions of A-to-I editing events in biological sequences. Using datasets derived from the public REDI portal, 300,000 editing sites were equally divided and used for learning sequence and structure motifs visualized by convolutional kernels. We explored the RNA editing pattern changes using information of positional and class enrichment of learned motifs in the three aforementioned classes. We demonstrated that these newly investigated approaches using large-scale RNA sequencing data offer excellent classification accuracy with a well-optimized convolutional neural network and recurrent neural network classifiers that obtained average area under curves of 0.960 and 0.962, respectively. The findings further decipher the principles underlying RNA editing events and will facilitate more effective RNA sequencing research.

INDEX TERMS A-to-I RNA editing, RNA-Seq, deep learning, convolutional neural networks, recurrent neural networks, structure motifs, repetitive elements

I. INTRODUCTION
RNA editing offers a dynamic and flexible means to alter the RNA sequence whereby a genetic message is changed from the corresponding DNA template by means of insertions, deletions, and base substitutions. These changes can directly affect protein expression and structure. RNA comprises a large number of subunits termed ribonucleotides comprising the following four bases: adenine (A), guanine (G), cytosine (C), and uracil (U). Under normal physiological conditions, an RNA sequence cannot exist stably in its primary structure. Therefore, to obtain a stable structure, it folds back itself to form secondary structures by hydrogen bonding between complementary bases. Generally, the RNA secondary structure is a set of stems stacked with base pairs. A base pair can be formed by three possible combinations including A-U, G-C, and G-U. The secondary information is vitally important as a bridge to understand RNA functions. Different ways to represent a RNA secondary structure have been devised. For instance, a string over alphabet A, C, G, U represents the primary structure of B, written as $h_1, ..., h_n$, and string S representing the secondary structure associated with string B can be written over the alphabet “(”, “), “”. Strings B and S must be compatible and parentheses in S have to be properly nested. When $(s_i, s_j)$ matches the
In particular, adenosine to inosine (A-to-I) editing is a post-transcriptional modification layer that is the most common type of RNA editing related to various biological processes, including neural function [1,2], cancer development [3,4], and the immune system [5]. Subsequently, the I residues can be decoded to guanosine (G) in a translation event that is mediated by double-strand, RNA-specific A deaminase that acts on the RNA (ADAR) family [6]. This process can expand the proteomic diversity of miRNA transcripts, which influences transcript stability, interaction with other primary RNA processes like splicing, the biogenesis and functions of microRNA and long non-coding RNA, and regulate gene expression. A-to-I editing is evolutionarily conserved and may trigger adaption in metazoans [7,8], and is essential for cellular functions that include amino acid changes in coding regions [9]; when unregulated, A-to-I editing contributes to various diseases. The abnormality of A-to-I editing sites shown in previous studies is closely related to mammals, especially in human diseases. These include central nervous system (CNS) diseases where abundant A-to-I editing sites in the CNS affect neural expression and receptors [10,11], CNS cancers [12], and cancer tissues [13]. Investigations seek to accurately identify RNA editing events.

Large-scale, genome-wide analyses of RNA editing sites have recently become feasible based on the availability of high-throughput RNA sequencing [14,15]. The development of these pipelines has made the study of RNA editing using RNA-seq datasets pervasive. Nonetheless, computational and technical challenges still make this task difficult [16]. Current methods are usually able to map short reads to a reference genome or transcriptome for DNA sequencing or RNA sequencing, respectively. These mapping flow charts are followed by removal of identical reads, filtering of low-quality reads, calling variants, and disposing of single nucleotide polymorphisms (SNPs) [17–19]. Only a few pipelines and computational tools are publicly available to analyze RNA editing events [20] that could map massive numbers of short reads to a references genome, due to the time involved. Furthermore, discrimination of RNA editing events from novel SNPs remains limited since DNA sequencing has not been routinely implemented in conjunction with RNA sequencing.

Deep learning has emerged as a successful paradigm for big data due to the technological advancements in terms of the lower cost of high computing architecture. Deep learning has achieved state-of-the-art success and has made significant contributions to arduous artificial intelligent tasks, such as computer vision, natural language processing, and machine translation. In particular, deep learning applications in bioinformatics have been successful in medicine and genomic medicine [49]. Convolutional neural networks (CNNs) [47], are one of the flexible and powerful deep learning models. CNNs have proved their worth in various applications including image detection and localization, and disease diagnosis via radiography. Designed to map image data to output variables, CNNs have proven efficient in the development of an internal representation of a two-dimensional image that allows the model to learn positions and scales of variants in the data. Characterization using convolutional filters, followed by rectified linear units enables sub-sampling as max-pooling and fully connected layers. Based on the very first CNNs, which helped propel the use of deep learning for several basic tasks, such as image classifications and predictions, Collobert et al. [29] firstly showed that CNNs could also be efficiently used for sequence analysis, especially for generic text. This attribute allows CNNs to be more generally adopted for other categories of data that have either a spatial or an ordered relationship in the time steps of a time series.

Recurrent neural networks (RNNs) were designed to exploit the sequential features from inputs with cyclic connection between building blocks as perceptions or long short-term memory units (LSTMs) [48]. RNNs are distinguished from feedforward networks by the feedback loop connected to past decisions, ingesting their own outputs moments after input moments. This is considered as memory. By adding memory to the network, RNNs can maintain the information of the sequence itself, while feedforward networks cannot. The sequential information is preserved in the hidden states of RNNs, which can span multiple time steps as it cascades forward to ameliorate the processing effect for each new example. In this manner, RNNs are able to find correlations between events separated by different moments. These correlations are termed long-term dependences. An event downstream in time depends upon one or more events that came before it, allowing it to share weight over time. For the techniques to be efficient on real problems, LSTM is applied in training using back-propagation through time, which can overcome the vanishing gradient problem. Instead of neurons, LSTMs have memory blocks connected into layers. Each block has components containing gates that manage the block’s state and output, making it smarter than a classical neuron. In turn, LSTMs can be used to address difficult sequence problems. RNNs outperform other architectures because they temporally handle sequence information, incorporating contextual information and localizing distortions from past inputs. This could help developing the promising sequence classification methods.

To the best of our knowledge, little is known of the use of deep learning architectures in simultaneous structural motif learning and sequence classification problems, including both CNNs and RNNs. Therefore, in this study, we introduced different settings of CNN models, which train the weights from scratch, using a very large available dataset, to overcome the analysis of massive RNA editing sites via sequence structure motifs. The aim was to discover its capability in sequence discrimination. The latter, a short-term memory network (LSTM) was the chosen RNN. We then compared the performances of those networks with chosen parameters to
explore their capabilities of learning features and the relations of RNA A-to-I editing site classes at different levels of taxonomy.

The rest of the work is organized as follows. Section II presents the related works done up to date in brief. Dataset descriptions, data preprocessing and CNN architectures for motif visualization and sequence classification are described in Section III. Section IV summarizes our experimental results using proposed methodologies. The work concludes in Section V with a short discussion and future works.

II. RELATED WORKS
One of the primary goals to understand the biological structure-function paradigm is sequence analysis. Traditionally established by sequence alignment methods such as BLAST [40] or FASTA [41], two main assumptions are motivated. Those consist of functional elements shared common sequence features and relative order of functional elements conserved between different sequences. There are few evidences suggesting that the order between elements would hold any important effect in gene regulation and expression in the case of cis-regulatory elements related sequence, as an examples of early alignment methods. They also still limit the application of alignments on very large sequence reads due to time computational complexity. As such, recently developed alignment-free methods [21] holds a promising approach to study regulatory genome. For feature extraction as spectral representation, several studies have shown significant research on DNA sequences [22,23] based on the performances of conventional machine learning algorithms.

Nonetheless, it is necessary to identify appropriate features to deal with specific tasks as a subsequent phase and this nowadays remains a very difficult step. The development of next generation sequencing technologies has greatly facilitated the RNA editing events identification. One of the first methods identified RNA editing using RNA sequences without the matching to genome reference [24]. The prediction tasks were then formulated to accurately predict constitutive RNA editing sites with new parameters, namely Hits Per Billion-mapped-bases and Potential SNP Score. Other tools were also introduced for related tasks. While GIREMI [25] utilized allelic linkage and built linear models to differentiate between RNA editing events and generic variants in single RNA-seq samples, RNAEdit2 [26] provided a friendly tool to identify these RNA-seq by developing clustering algorithms to find out those editing regions. Although these techniques greatly facilitated RNA detection on public transcriptomic sequencing datasets without referring to DNA sequencing data, it was cumbersome and time-consuming when filtering steps were done based on prior knowledge and public genome annotations such as Alu repeats, pseudogenes and genomic duplications. In addition, arbitrary combinations of filtering steps with insufficient genomic annotations and high demand of domain specific knowledge might result in wrong detection of different RNA editing candidates.

Therefore, a fast, high performance, the computationally efficient and user-friendly RED-ML tool [27] served the RNA editing detection. By adopting machine learning techniques, RED-ML could automatically and optimally combine different sources of sequence information and detect RNA editing sites in a robust manner. While a simple BAM file is the input, the output contained detected RNA editing sites, as well as confidence score of downstream filtering. They carefully designed validation experiments and extensive comparison to show the RED-ML efficiency under different conditions to excellently detect novel RNA editing events without relying on curated RNA databases. Another deep learning-based method called DeepRed [28], in contrast, could identify RNA editing sites by directly learning and finalizing essential features from surrounding primitive sequence of RNA editing candidates without requiring prior knowledge and filtering steps. DeepRed manifested good generalization capability by achieving 97.9% of AUC. They also discovered the RNA editing pattern changes during early embryogenesis in human and evolutionary patterns of RNA editing events in Drosophila species. As a new finding, they discovered the stage-specific change patterns of RNA editing events at 8-cell, morula and blastocyst stages during evolutionary conservation of RNA editing between close lineages and human early embryogenesis. RES-Scanner [30], called RES: RNA-editing site, in addition, was introduced as flexible and effective software package written by Perl programming language to develop genome-wide identification and annotations of RNA editing events. It was also designed to address sequence read mapping, homozygous genotype calling, matching RNA-seq and DNA-seq for any species.

However, there were no such works to interpret structure and enrichment motifs, as well as to classify RNA A-to-I editing locations from those available tools. In our works, the three-categories classification task would be targeted by superior performances of CNNs and RNNs to know if each editing region is located in ALU-residing in Alu repetitive elements, REP-located in non-Alu repetitive elements or NONREP-placed in non-repetitive regions based on the information of motif pattern enrichments. The results can greatly trigger further researches on RNA editing mechanisms.

III. MATERIALS AND METHODOLOGIES

A. DATASETS
Derived from REDIportal [28], known as the largest collection of high-throughput genomic data to study gene expression in 30 distinguished tissues, where is a freely available database

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>RNA EDITING DATASET DIVISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category</td>
<td>ALU</td>
</tr>
<tr>
<td>Training (60%)</td>
<td>60,070</td>
</tr>
<tr>
<td>Validation (20%)</td>
<td>19,990</td>
</tr>
<tr>
<td>Testing (20%)</td>
<td>19,940</td>
</tr>
<tr>
<td>Total (sequence reads)</td>
<td>100,000</td>
</tr>
</tbody>
</table>
collecting over 4.5 million RNA A-to-I editing events from 2642 RNA-seq experiments in 55 body sites of 150 healthy individuals from GTEx project, we intensively sampled 300,000 editing sites by searching genomic region, gene name and other relevant information as the tissue of origin. The query results could be then shown in sortable and downloadable tables in which the major characteristics of individual RNA editing site are reported. In fact, we conveniently obtained RNA editing events datasets from REDIportal without primary filtering, alignment or calling variants steps. GTEs datasets are downloaded from Genotypes and Phenotypes (dbGaP) in SRA format and converted to standard FASTQ files by fastq-dump tools [42]. FASTQ files are next checked its quality by FASTQC and align onto the reference human genome by STAR tool [43]. REDIportal finally interrogates multiple read alignments of very large collection of known RNA editing sites from ATLAS repository [44] and RADAR database [45]. After collecting equivalent number of sequence reads of three desired positions, including ALU, REP-NON-ALU and NON-REP, we split datasets into training, validation and testing sets before putting them to our CNNs (Table I).

B. PREPROCESSING

For easier training, learning structure-sequence motifs, and evaluating the efficiency of CNNs in sequence classification, the input is carefully handled. Raw strings from REDIportal [28] are in FASTA and additional structural formats. To visualize the motif patterns, one-hot encoding input conversion is necessarily considered [50]. While sequence-only files including four alphabets (A, C, G, and U) are provided, a tuple following special characters “()[]{<,,}” can be used for sequence-structure files. Additional handcrafted features or optionally position-weight matrices for structure data can be added. The provided alphabets have to match the length of the FASTA files. Characters that do not comprise any of the aforementioned alphabets would be randomly replaced with an alphabet character. We ensured that each string contained 300 characters so that the sequence and structure data in all files had the same length.

Once first-preprocessed sequences were ready, we automatically converted them into one-hot encoded matrices via RNAfold binary to annotate secondary structure with structural contexts from structure data. In particular, given dot-bracket strings, we annotated every character as either ‘H’ – hairpin, ‘S’ – stem, ‘I’ – internal bulge or ‘M’ – multi lop. Together, a first-preprocessed RNA sequences with dot-bracket structure data would be converted to FASTA files with secondary structure annotations.

C. PROPOSED DEEP LEARNING ARCHITECTURES

1) CONVOLUTIONAL NEURAL NETWORK (CNN)

In order to visualize and interpret sequence motifs, we handled both RNA sequence reads and corresponding secondary structure strings by encoding them into a new matrix with extended matrices (Fig. 1, the dashed line pointer). More precisely, four nucleotide RNA characters are combined with four annotated secondary alphabets to generate a new matrix input comprised of sixteen arbitrary characters from A to P. The new matrix string is then fed into the first convolutional layer of our CNN that is calculated by L one-dimensional convolutions with input x, the kernel vectors $w^l$ with the size $2n + 1$:

$$q^l(i) = \sum_{u=-n}^{n} w^l(u)x(i - u)$$  \hspace{1cm} (1)

where $q^l(i)$ is the element $i$ of $l$-th output vector while $w^l(u)$ is the component $u$ of $l$-th kernel vector. The nonlinear function $g$ is then applied by adding a bias term $b^l$ is denoted as:

$$h^l(i) = g(q^l(i) + b^l)$$  \hspace{1cm} (2)

Convolutional layers are used to extract subsequences included dominated motif features and visualize motif patterns by decoding into the two original strings via convolutional kernels. This facilitates Position Weight Matrix (PWMs) to be computed and points out where these subsequences indicate positional enrichments. By definition $\sum_{i=1}^{4} w_{k,i} = 1$, where $w_{k,i}$ notes the probability of observing nucleotides $i \in \{1,2,3,4\}$, representing $\{A,C,G,T\}$ in the position $k \in \{1,\ldots,l\}$ of RNA editing sites. Then, the position score $S(W, s)$ for a string $s$ and the length $l$ is defined by:

$$S(W, s) = \sum_{k=1}^{l} \ln \left( \frac{w_{k,s_k}}{s_k} \right)$$  \hspace{1cm} (3)

where $s_k$ is the $k$-th base of $s$. Motif features significantly correspond to kernels that make RNA A-to-I editing sites classification results more accurate.

In contrast, the typical end-to-end CNN frameworks [33,34], which have been inclusively applied for sentence classification, are used for multi-class discrimination tasks. RNA sequences and the secondary structure alphabets are directly put into our CNN consisting of convolutional layers followed by max-pooling layers (Fig. 1, solid line pointer). The dropout layer to regularize our model is interspersed between the convolutional and max pooling layers followed by a dense layer. The output vector is finally fed to a softmax function as the last layer that maps the hidden features from previous layers to the class label space $C \in \{0,1,2\}$. By using an automated grid search, our CNN models can be tuned via the appropriate changes of hyper-parameters values, such as the number of convolutional layers, number of kernels, and dropout ratios (Table II). We observed that the higher number of convolutional layers with more kernel sizes and better classification accuracy can be obtained. The findings indicate that the deep convolutional network successfully outperforms...
traditional feedforward networks in dealing with biological sequence problems.

2) RECURRENT NEURAL NETWORK (RNN)

The primary goal of RNNs is to present the variable-length text for specific-task text classification as a fixed-length vector. A RNN model consists of a projection layers that can map words, sub-word units to vector representations trained beforehand in unsupervised manner, and their combination with parameters to formulate the complete RNNs. This idea has been presented by previous works such as Neural Bag-of-Words (NBOW) model [31] or recursive neural network (RecNN) [32]. These models took inputs of the embedding of words in the full text sequence and next summarized its meaning based on a fixed length vector representation. Generally, a typical RNN owns an internal state $h_t$ summarizing the sequence seen until the usage of both the previous time step $(t - 1)$ and new input $x_t$ is denoted as:

$$ h_t = \sigma(W_h x_t + U_h h_{t-1} + b_h) \quad (4) $$

$$ y_t = \sigma(W_y h_t + b_y) \quad (5) $$

where $W_h$ and $U_h$ are weight matrices for input and internal state while $W_y$ is the weight matrix of output produced by the internal states and $b$ is the bias vector.

Having built our RNN model on that original idea, we adjusted different settings with different layer arrangements to see which setting is the most appropriate for our study. We then finalized our RNN comprising six-layer network (Fig. 2). The RNN starts with the inputs similar to the input forms of our CNN are one-hot encoding vectors. The next layers include an embedding layer, followed by a max pooling layer of size 2 to reduce computational cost for the following layer and simultaneously bestow translational invariance to the network.
A recurrent layer is an LSTM that processes the features after generating an output vector of size 20 for every time step. Another sub-sampling layer is then followed to feed learned features to the two fully connected layers. The final softmax layer, which acts as a probability classifier to differentiate RNA editing classes, is finally involved. Both CNN and RNN use a similar softmax formulation composed of $K$ units, where $K$ is the number of classes. Computing the probability for each class is denoted by the following formula:

$$\text{softmax}_k(x) = \frac{e^{W_kx + b_k}}{\sum_{k=1}^{K} e^{W_kx + b_k}},$$

(6)

where $W_k$ is the weight matrix that connects $l$-th unit to the previous layer, the output of the previous layer $x$ and the bias for the $l$-th $b_l$.

Deep learning framework is performed by a system with NVIDIA GTX 1070Ti and Ubuntu 16.04 and TensorFlow 1.9 with CUDA 9.0/cuDNN 7.5 dependencies. 10-fold cross validation is also applied that easily evaluates the most appropriate parameters for our both models. The training time costs the average of 276.08 minutes for three time experiments.

IV. EXPERIMENTAL RESULTS

A. STRUCTURE MOTIF VISUALIZATION

As mentioned earlier, in genomic sequences, the genetic information of A-to-I RNA editing sites is strongly influenced by local sequential patterns known as motifs. Such structure motifs can be visualized as the temporal equivalent of spatial patterns in natural images, which is what our CNNs is able to automatically learn, extract dominated features from these motifs and then evaluate which parts of RNA sequence were holding the most influential information for further classification problems. We first considered a motif as an abstract description of a set of aligned sequence with length $L$ and the alphabet $A$ including four nucleotides. The relative frequency symbol $s \in A$ at position $l \in [1, L]$ associated with the probability $p_{l,a}$. In case of visualizing two motifs simultaneously, $p_{l,a}$ is used for the first motif while analogously $q_{l,a}$ is used for the second motif. The well-known notation and sequence logo is computed to visualize a motif by the symbol stack at each position $[51]$. The height of the stack at position $l$ is denoted as $H_l$ while the height of a symbol $s$ within this stack is noted as $H_{l,a}$. The sequence logo, $H_l$ and $H_{l,a}$, are defined by:

$$H_l = \log_2(|A|) = \sum p_{l,a} \cdot \log_2(p_{l,a}) \quad (a \in A)$$

(7)$$H_{l,a} = p_{l,a} \cdot H_l$$

(8)

These two equations state the height of the stack at position $l$ reflecting the degree of conservation and the height of each symbol $a$ is proportional to its frequency at position $l$. Thus, the sequence logo is an intuitive visualization of both conserved motif positions and abundant bases.

Next, we seek to visualize each position’s influence of subsequences on the prediction. Precisely, we did a pointwise multiplication with the one-hot encoded sequence for both actual nucleotide characters and secondary structure

![Fig. 3a. An example of motif visualization of a RNA sequence and its corresponding secondary structure sequence (left) and activation values representing the dominated features among three RNA editing classes at convolutional kernel 11.](image)

![Fig. 3b. Detailed histograms showing the positional enrichments of three RNA editing classes at kernel 11.](image)
sequences to get derivative values. The output of the convolutional layer is called activations that are results of sliding with the kernel over an input sequence and one position at a time. The maximum activation shows which subsequence of the kernel is most similar. When the maximum activation is above a certain threshold, the corresponding subsequence of the length of the kernel from the input starting at that specific position is extracted. Doing this way for a number of subsequences with all of the same length, PWM can be computed. Hence, the output score may indicate the influence of these characters at each position. Next, the element-wise magnitude from derivative vectors can visualize how important each character is regardless to the derivative direction. It points out which nucleotides change the least but affect the class score the most. In our results, we plotted the position of the maximum activation as a histogram that is able to show exactly where our motifs are located. By doing this procedure for every class, it gives class motif enrichments.

Fig. 4a. Information of motif visualization of a RNA sequence and its corresponding secondary structure sequence, the activation values and histograms representing the dominated features among three RNA editing classes at convolutional kernel 12 of Class_2—REP class are all higher than other two classes.

Fig. 4b. Information of motif visualization of a RNA sequence and its corresponding secondary structure sequence, the activation values and histograms representing the dominated features among three RNA editing classes at convolutional kernel 14 of Class_3—NON-REP class are all higher than other
Collectively, information concerning the visualization of non-REP (Class_3) has higher values to present the class compared to other class at kernel 12 (Fig. 4a). Similarly, Class activation value and the positional frequency is much higher (Class_2) shows stronger enrichments according to the strength of the sequence effects on the specific regions. Moreover, the activation values of kernel 11 of ALU class tend to repetitive regions, tend to have stronger structural motif patterns than other two classes, including REP and NON-REP classes belonging to non-ALU repetitive regions and non-repetitive regions, respectively. RNN-LSTM, in comparison, performs better than CNNs in terms of both AUC score of 0.962 and 93.53% of classification accuracy, although in terms of training time it is unfavorable, since it takes twice as long as CNNs for the same number of iterations and mini-batch sizes. We also compared other evaluation values for its better performances compared to CNNs (Table III).

**B. RNA EDITING SITES CLASSIFICATION ANALYSIS**

Table II summarizes the means of area under the ROC curve (scores) for three different setting CNNs and RNN-LSTM. In general, we achieve very good AUC scores and precision accuracies in classifying multi-RNA editing categories. This validates our hypothesis that all of three different settings of CNNs can possibly trigger local motif patterns from sequence reads of RNA editing sites. Moreover, as expected, the well-optimized CNN_3 with largest number of convolutional layers and well-fitted hyper-parameters significantly outperforms the other two simpler CNNs. It achieves 0.960 of AUC score and 91.31% of 3-class discrimination accuracy. In addition, to more precisely evaluate the classification capability of CNN_3, we showed the different evaluation values obtained by assessment of specific numbers of testing datasets (Fig. 5 and Table III). The results demonstrate that the ALU class has higher values of all the evaluation strategies. It specifically means the ALU class, which represents A-to-I RNA editing sites that belong to repetitive regions, has much higher structural motif patterns than other two classes, including REP and NON-REP classes belonging to non-ALU repetitive regions and non-repetitive regions, respectively. RNN-LSTM, in comparison, performs better than CNNs in terms of both AUC score of 0.962 and 93.53% of classification accuracy, although in terms of training time it is unfavorable, since it takes twice as long as CNNs for the same number of iterations and mini-batch sizes. We also compared other evaluation values for its better performances compared to CNNs (Table III).

![Fig. 3a](image)

Fig. 3a illustrates an example of the motif interpretation by visualizing the structure motifs of a random RNA sequence and its corresponding secondary structure sequence via convolutional kernel 11. The score of 0.390 indicates the strength of the sequence effects on the specific regions. Moreover, the activation values of kernel 11 of ALU class (Class 0) is higher than the REP and NON-REP class, as shown in Fig. 3b. We also plotted histograms of that kernel to visualize the positional enrichment presenting that sequence positions at which subsequences corresponding to the kernel activations higher than a desired threshold are extracted. ALU class enrichment is much more distributed in the interval [0-200] of sequence regions compared to the other two classes. Collectively, information concerning the visualization of motifs, higher values of activation kernels, and stronger distributions of positional enrichment of that convolutional kernels result in easier prediction for the ALU class.

At different kernels, for instance, class REP-NON-ALU (Class 2) shows stronger enrichments according to the activation value and the positional frequency is much higher compared to other class at kernel 12 (Fig. 4a). Similarly, Class Non-REP (Class 3) has higher values to present the class enrichments at kernel 14 (Fig. 4b). These visualizations are highly correlated with the expression of motif patterns and classification of multiple categories of RNA editing sites.

**Table II**

<table>
<thead>
<tr>
<th>Model</th>
<th>Conv. Layers</th>
<th>Conv. Kernel Size</th>
<th>Conv. Filter Size</th>
<th>AUC score</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small CNN-CNN_1</td>
<td>4</td>
<td>20</td>
<td>9,5,3</td>
<td>0.935</td>
<td>77.21</td>
</tr>
<tr>
<td>Medium CNN-CNN_2</td>
<td>9</td>
<td>20</td>
<td>9,5,3,3,3</td>
<td>0.942</td>
<td>83.54</td>
</tr>
<tr>
<td>Large CNN-CNN_3</td>
<td>17</td>
<td>64</td>
<td>9,5,3,3,3</td>
<td>0.960</td>
<td>91.32</td>
</tr>
<tr>
<td>RNN-LSTM</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.962</td>
<td>93.53</td>
</tr>
</tbody>
</table>

**Table III**

<table>
<thead>
<tr>
<th>Model</th>
<th>Category</th>
<th>No. of testing samples</th>
<th>Recall</th>
<th>F1-score</th>
<th>AUC score</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN_3</td>
<td>Class_0 - ALU</td>
<td>19940</td>
<td>0.964</td>
<td>0.972</td>
<td>0.999</td>
<td>98.04</td>
</tr>
<tr>
<td></td>
<td>Class_1 - REP</td>
<td>19900</td>
<td>0.712</td>
<td>0.787</td>
<td>0.936</td>
<td>90.90</td>
</tr>
<tr>
<td></td>
<td>Class_2 - NON-REP</td>
<td>20160</td>
<td>0.931</td>
<td>0.844</td>
<td>0.946</td>
<td>86.03</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>60000</td>
<td>0.869</td>
<td>0.868</td>
<td>0.960</td>
<td>91.32</td>
</tr>
<tr>
<td>RNN-LSTM</td>
<td>Class_0 - ALU</td>
<td>19940</td>
<td>0.972</td>
<td>0.973</td>
<td>0.987</td>
<td>99.70</td>
</tr>
<tr>
<td></td>
<td>Class_1 - REP</td>
<td>19900</td>
<td>0.724</td>
<td>0.808</td>
<td>0.938</td>
<td>91.80</td>
</tr>
<tr>
<td></td>
<td>Class_2 - NON-REP</td>
<td>20160</td>
<td>0.951</td>
<td>0.862</td>
<td>0.961</td>
<td>89.10</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td>60000</td>
<td>0.882</td>
<td>0.881</td>
<td>0.962</td>
<td>93.53</td>
</tr>
</tbody>
</table>
machine learning algorithms for sequence classifications and modeling frameworks for sequences. The five different biological datasets utilized by the authors could be divided into two broad categories: nucleotide sequences having an alphabet of four characters assigned a class label of either intron or exon, and amino acid sequences having an alphabet of 20 letters that belonged to either coding regions or non-coding regions. Their findings showed that the Support Vector Machine (SVM) was able to achieve higher sequence differentiation accuracies compared to k-nearest neighbors (kNN) and Markov model approaches. In addition, Wang et al. [36] identified the top 23 most informative features from differentially expressed genes (DEGs) using different feature selection algorithms with machine learning classification methods. They found that the model based on InfoGain selection algorithms combing Logistic Regression classifier was most powerful in predicting DEGs. The authors reported promising results by collecting 468 features of two categories (histone acetylation over different gene segments and histone peak-associated features) and accessing the performances of two main classifiers (logistic regression and random forest). However, as shown in Table IV, our deep learning approaches outperform the abovementioned conventional methods both in terms of the amount of data and classification results.

Regarding deep learning approaches, Jurtz et al. [38] illustrated how their architectures, including CNN and CNN-LSTM neural networks, could relatively and easily be designed for biological problems that consisted of prediction of subcellular localization, the binding of peptides to MHC Class II molecules and protein secondary structure prediction. Similarly, the CNN model was also used for enhancer identification from DNA sequences in Cohn et al. [52]. By sampling 10K positive and 10K negative 500bp long DNA sequences, they achieved a set of biologically meaningful motifs – unique to enhancers and a high classification accuracy based on the combination of two strategies: first, training on enhancer and non-enhancer sequences to identify short (1-4bp) low complexity motifs; then, replacing the negative set by adversarial k-order random shuffles from enhancer sequences. Khawaldeh et al. [53] also applied CNNs to classify each DNA sequence to which taxonomy of classes, orders, family, etc. They encoded DNA sequences by creating seven unique labels according to the order name that was expected to help CNN perform better. Their results similarly showed a very high potential of incorporating CNN model and DNA encoding. Another work proposed the pipeline to stimulate both SG and AMP short-reads from 16S full-length sequences for bacteria taxonomic classification of metagenomics data [54]. They adopted a k-mer representation technique to map sequences as vectors before feeding them into CNN and DBN. In specific, in the genus level, both CNN and DBN reached to 91.3% of accuracy with AMP short-reads, whereas Bayesian classifier obtained 93.8% with the same data.

Regarding the closer studies to ours, Lee and Nguyen [39] explored how well Gated Recurrent Neural Networks (GRU), compared to SVM, CNN and bidirectional LSTM-based architectures, could represent biological function via examination of only raw sequences. They learned dense vector representations for a large corpus of protein sequences and their annotated protein families using co-occurrence statistics of short fragments. By this representation, their proposed classifiers could train to identify 589 protein family classes and 317,460 associated sequences. Using different settings of experiments for the task of protein family identification, they demonstrated that LSTM model gave a superior performance to their CNN and tuned SVM baselines on the same Uniprot dataset with 0.925, 0.897 and 0.878 F1-scores, respectively. In comparison, it shows that our proposed CNNs and RNN via F1-score values are slightly lower than their studies. On the other hand, we earned a competitive result compared to a customized CNN and RNN with no steps of sequence processing from Bosco et al. [35] who studied on the three-class Phylum (coarsest grained) taxonomic rank with only selected 3,000 sequences to be discriminated. They proposed LSTM that exploited the nucleotides positions of sequences, in comparison with a basic CNN. Their results also showed that LSTM worked better both in terms of performances and
training times with 99.50% and 98.20% accuracies, respectively. However, taking into consideration the accuracy results of the state of the art, their dataset is insufficient to accurately evaluate their performances compared to our study.

V. SUMMARY AND CONCLUSIONS
Classification of RNA editing events is very important for research on post-transcriptional regulation. Large-scale sequencing efforts are being investigated to permit the accurate identification. However, this approach is very expensive and time-consuming. In this study, we adopted highly effective and applicable CNNs that are able to interpret meaningful sequence-structure motifs via convolutional kernels. We compared different settings of CNNs to obtain the optimal CNN for our RNA A-to-I editing datasets, resulting in 91.32% precision accuracy. RNN outperformed, with 93.53% verification rate of RNA editing regions, and thus also successfully discriminated three RNA editing categories. Nevertheless, a targeted analysis of RNA sequences from the same individual remains a challenge. Relatively high editing levels from the chosen RNA editing datasets limits the detection of a variety of editing levels such that the non-editing events can be mistaken as editing events if the wide range of RNA editing levels is not presented. In the future, we intend to retrain our models with comprehensive training datasets consisting of more accurate and larger RNA editing scales. With the improved knowledge on RNA editing mechanisms, we will be able to design more features and parameters needed in deep learning architectures. In addition, classification tasks on such similar datasets of DNA and protein sequences need to be addressed.

REFERENCES


