RNA Splice Sites Classification Using Convolutional Neural Network Models

Thanyathorn Thanapattheerakul
School of Information Technology, King Mongkut's University of Technology Thonburi,
Bangkok, Thailand
thanyathorn.tha@mail.kmutt.ac.th

Worrawat Engchuan
The Centre for Applied Genomics, Genetics and Genome Biology, The Hospital for Sick Children,
Toronto, Ontario, Canada
worrawat.engchuan@sickkids.ca

Daniele Merico
Molecular Diagnostics, Deep Genomics,
Toronto, Ontario, Canada
daniele.merico@gmail.com

Narumol Doungpan
Faculty of Engineering, King Mongkut's University of Technology Thonburi,
Bangkok, Thailand
narumol.dou@sit.kmutt.ac.th

Kiyota Hashimoto
Faculty of Technology and Environment, Prince of Songkla University,
Phuket, Thailand
hash@reasoning.jp

Jonathan H. Chan
School of Information Technology, King Mongkut's University of Technology Thonburi,
Bangkok, Thailand
jonathan@sit.kmutt.ac.th

***Abstract*—RNA splicing refers to the elimination of non-coding region on transcribed pre-messenger ribonucleic acid (RNA). Identifying splicing site is an essential step which can be used to gain novel insights of alternative splicing as well as splicing defects, potentially cause malfunction of protein resulting from mutations at splice site. In this work, we propose a data preprocessing step applying to RNA sequences and the models leveraging Convolutional Neural Network (CNN). The preprocessing step includes reducing sequence length into 40 nucleotides. CNN models recognize splice sites on RNA sequences. Our proposed models output the promising results which increase F1-score (nearly 20%) comparing with recent alternative approaches when testing on GWH dataset.**

Keywords—RNA splice sites, Bio-inspired computing, Computational biology, Deep learning predictive models in bioinformatics

I. Introduction

In eukaryotic cell, biological information flows from deoxyribonucleic acid (DNA) to ribonucleic acid (RNA) to protein. DNA is a sequence of four types of nucleotides: adenine (A), guanine (G), cytosine (C), and thymine (T). RNA has the same components as DNA, except Uracil (U) instead of thymine. To transfer genetic information to synthesize protein, there are three main steps: transcription, splicing and translation.

During transcription, DNA is copied into a precursor messenger RNA (precursor-mRNA or pre-mRNA). Before the pre-mRNA becomes a mature messenger RNA (mRNA), which directs the synthesis of Protein in translation process, some regions on pre-mRNA are removed by splicing. The non-coding regions or 'introns' intervene between 'exons,' which is the protein-coding regions. The boundary between an exon and intron is referred to the 'splice site' (junction). The characteristic of exons and introns is having a dinucleotides GT or donor and AG or acceptor at the boundaries [1].

Some RNA molecules have the capability to splice themselves, but some splice alternatively. The splicing research [2] discovered that alternative patterns of splicing within a single precursor-mRNA at different junctions could yield in a variety of mature mRNAs. Most of the alternative splicing is caused by a mutation of a splice site, which can reduce spliceosome binding specificity of that splice site or completely make it loss of function. The alternative splicing can produce different functional proteins, which could lead to causing abnormal states in human [3].

Many studies have proposed models to recognize the splice sites to reveal which splice sites contain a mutation that may cause a splicing error. One common method to recognize binding sites in motif sequences is called Position-Weight-Matrix (PWM). PWM recognizes by transforming sequence data into a probability matrix [4]. It can be used to recognize simple sequence structures; however, growing evidence indicates that sequence specificities can be more accurately captured by more complex techniques [5-10].

Recently, machine learning (ML) and deep learning (DL) has become popular. One well-known and effective ML technique is support vector machine (SVM). Degroeve et al. have proposed a publicly splice site prediction tool called SpliceMachine [5]. They employed a linear support vector machine (LSVM) as a linear classifier to predict the boundary between an actual and pseudo splice sites. Similarly, Sonnenberg et al. have leveraged SVM and employed weighted kernel to predict the splice sites [6]. The more recent studies have adopted DL techniques. DL is one of neural networks (NNs) techniques, inspired by human brain neurons. In bioinformatics, DL techniques have been applied to many problems, including DNA-binding sites recognition [11]. In 2015, Lee and Yoon proposed a deep belief network-based methodology, which applied Boltzmann machines for prediction of splice sites [1]. Their method showed the potential of DL techniques by achieving better results than other alternative methods (e.g. SVM and Hidden Markov Model (HMM)). DeepBind [7] and DeepSEA [8] are state-of-the-art approaches that also proposed DL-based models for predicting sequence specificities and effects of non-coding genomic variants, respectively. They found that DL-based models compete favorably with Convolution Neural Networks (CNNs). Recently, SpliceRover has been introduced for RNA splice sites recognition by using CNNs and results showed promising improvement [10].

Back in February 2018, the School of Information Technology, King Mongkut's University of Technology Thonburi (KMUTT) in Thailand held the First Deep Learning and Artificial Intelligence Winter School (DLAI1) [12]. The event focused on the state-of-the-art technologies in artificial intelligence, especially machine learning and

deep learning. There were many well-known professors and industrial people who have been working on these areas. In addition, a competition track giving a financial data, was included. This December, the second of DLAI will be jointly organized with two international conferences. The competition track will use a genomic data, particularly RNA sequences with actual and pseudo splice sites, supported by Deep Genomics [13].

In this paper, we propose the deep learning models to recognize RNA splice sites by leveraging CNNs. These models are able to recognize actual splice sites on 40 nucleotides of sequence length.

In the next section, we will describe about the data and methodology. The results and discussion will be followed. We end the paper with conclusion and future works.

II. Experimental and Computational Details

A. Data

We use a dataset published during the First Deep Learning and Artificial Intelligence Winter School (DLAI1) [12]. The dataset was collected, and minimum preprocessing was completed. It is also planned to be public for a competition to be held in this year. The dataset consists of two data files:

- *Positive Case Data.* The positive case data is used to extract splice sites (donor and acceptor). The donor and acceptor files contain 223,143 and 220,034 sequences of 40 nucleotides, respectively. Each sequence has a splice site in the center, *GT* for donor sequences and *AG* for acceptor sequences. The splice site will be extracted and used as cases for model training.
- *Negative Control Data.* The negative control data is obtained by extracted sequences of the same size of splice sites mentioned above. Those sequences will be from highly conserved dinucleotide regions that are not a splice site. Sequences of 40 nucleotides with core dinucleotide located at the same position found in actual splice site sequences.

B. Methodology

To achieve neural network models for recognizing RNA splice sites, our methodology consists of three main steps as shown in Fig. 1.

Fig. 1. An Overall Workflow

1) Data Preprocessing. Normally, CNNs receive a vector of numbers representing pixels in an image. Therefore, splice site data are processed to an appropriate format as an input of models. To achieve that, raw RNA sequences are converted into a binary matrix using a one-hot encoding, A, C, G, T correspond to the vector [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0] and [0, 0, 0, 1] respectively. Therefore, the data will be represented as an image data of resolution $N \times 4$, where N is the length of an RNA sequence. Fig. 2. shows an RNA sequence and a binary matrix format.

2) Modeling. To detect patterns in RNA sequences, we employ neural networks and convolutional neural networks, including shallow and deep architectures, to our models.

Neural networks or NNs are designed for representation of high-level abstraction in data. NNs consist of different layers, which hold the number of neurons or nodes. Each neuron has different parameters or weights. Typically, a network consists of an input layer, hidden layers and an output layer. The input layer is propagated the data through the network, yielding intermediate results by using activation functions for each hidden layer. The output layer results a final prediction. For the complex models that are able to represent non-linearity, non-linear functions are applied to activate neurons in each layer. For example, Rectified Linear Unit (ReLU) [14], softmax [15] and so on. ReLU has been applied to most recent deep neural networks for each neural in hidden layers. It is the simplest non-linear function, which output 0 if the input is less than or equal 0, and raw output otherwise. In classification problems, softmax is typically used as an activation function in the last layer to generate a probability of each class. In supervised learning, NNs learn from the annotated training data by adjusting the weights based on a loss function. The loss function represents the difference between the predictions of the network and the annotated labels. However, a hidden layer of a NN has individual, independent weights for every position in the input it receives. This implies that it is not possible to learn to look for a particular pattern over a whole sequence. For that purpose, Convolutional Neural Networks (CNNs) were introduced. These are a kind of DL, which is NNs with more hidden layers, convolutional layers in this case. Each of these layers has a number of filters, sliding over the sequence and detecting patterns. Here, weights are stored within a filter to be shared over different positions.

In this paper, we propose DL-based models for recognizing actual and pseudo RNA splice sites. We experiment with several neural network architectures, which are two CNN architectures and one shallow neural network. CNN architectures consist of a number of alternating convolutional layers, dropout and max-pooling layers. Dropout and max-pooling layers have been employed for preventing over-fitting by deactivating some neurons in each layer and reducing dimensionality of an input, respectively. We employ ReLU as an activation function for every neuron. Next, they are followed by a fully-connected layer to conclude with softmax classifier [15]. The classifier results a probability for class 0 and 1. The neural network model consists of one hidden layer, followed by dropout. It is ended with one fully-connected layer with softmax classifier.

To construct our models, we firstly reproduce SpliceRover models [10], however, we find that they are not fit to our input. Therefore, we modify architectures and fine tuning hyperparameters. The exact parameters and hyperparameters of our topology will be described in the next section. Testing process is then conducted, and all results are described in next section.

3) Prediction. To obtain models for RNA splice sites prediction, we start with training process. Training process

is conducted on a set of RNA sequences that have been converted to the binary matrix. The raw data is then split by 5-folds cross-validation. 20% of a set of training data is assigned to be a set of validation data.

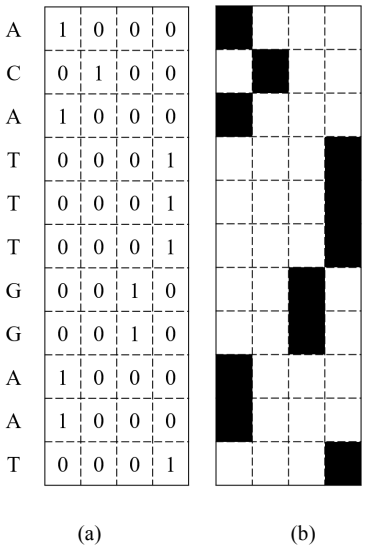

Fig. 2. Binary Matrix of a Sequence (a) an RNA sequence transformed as a binary matrix and (b) the RNA sequence representing as an image.

C. Computational Setup

All experiments are conducted on a Google Colaboratory with Python 3 and GPU hardware accelerator. We make use of Pandas and NumPy for data preparation and representation. TensorFlow and Keras are used for constructing models, training and testing.

III. Results and Discussion

To evaluate the effectiveness of the proposed models, we compare our approach to previous techniques. The benchmark methods we use are Lee and Yoon's work and SpliceRover.

Lee and Yoon conduct test on Genome Wide dataset for Human (GWH, [6]), containing two types of datasets: donor and acceptor. Each sequence in donor and acceptor is 398 nucleotides of length. All sequences have dinucleotides, GT and AG, in the middle. Each donor and acceptor sequence have GT in positions 200 and 201, and AG in positions of 198 and 199. GWH dataset is a substantial imbalanced data between two classes in both donor and acceptor. The donor has 1,565,360 sequences where only 5.14% are positive. Similarly, 5.45% of 1,484,845 acceptor sequences have actual splice sites. Lee and Yoon evaluate their methods and report by F1-score. F1-score is one of evaluation metrics considering as the harmonic mean of precision and recall. It reaches its best value at 1 and worst otherwise. SpliceRover, CNNs-based model, is tested on the same dataset and evaluated by F1-score. To compare our approach to these previous methods, we test our models on GWH dataset and report F1-score as follows.

In Table 1., we present our models' architectures and parameters. We have three architectures: two are CNNs-based and another is NNs-based. Two CNNs-based architectures are different in terms of depth or a number of convolutional, dropout and max-pooling layers. The first one is deeper than another one. The 2nd CNNs-based does not have a max-pooling layer. The hypothesis behind is because the sequence length is pretty short comparing to previous approaches and it should not be decreased by applying max-pooling layer, which we hypothesize that this architecture would give better evaluation metrics than the 1st CNNs-based. The NNs-based has an uncomplicated architecture. Hyperparameters we use are shown in Table 2. SpliceRover's hyperparameters are used as a baseline at first then we fine-tune to improve the performance as the second one.

TABLE I. Proposed Architectures and Parameters

Name	**Architectures**	
	Layers	***Details***
1st CNNs	conv2D layer 1	70 filters of size (3,4)
	dropout layer 1	p = 0.2
	conv2D layer 2	100 filters of size (3,1)
	dropout layer 2	p = 0.2
	conv2D layer 3	100 filters of size (3,1)
	maxpool layer 1	pool size (2,1)
	dropout layer 3	p = 0.2
	conv2D layer 4	200 filters of size (3,1)
	maxpool layer 2	pool size (2,1)
	dropout layer 4	p = 0.2
	dense layer 1	512 neurons
	dropout layer 5	p = 0.2
	softmax layer	2 outputs
2nd CNNs	conv2D layer 1	70 filters of size (5,4)
	dropout layer 1	p = 0.2
	conv2D layer 2	100 filters of size (3,1)
	dropout layer 2	p = 0.2
	dense layer 1	512 neurons
	dropout layer 3	p = 0.2
	softmax layer	2 outputs
NNs	dense layer 1	128 neurons
	dropout layer 1	p = 0.2
	softmax layer	2 outputs

TABLE II. Hyperparameters of Our Proposed Models

Set #	**Optimizer**	**Loss function**	**Epoch**	**Batch size**	**Start learning rate**	**Learning decay**
1	SGD with Nesterov momentum 0.9	Categorical cross-entropy	30	64	0.05	5
2	RMSProp	Binary cross-entropy	30	64	0.001	0

The results from our proposed models when perform on GWH dataset are shown in Table 3. As can be seen, the DL-based models tend to give better results than NNs-based. The 1st CNNs-based model with our fine-tuning hyperparameters gives better results, in both donor and acceptor, than another set. The 2nd CNNs-based model shows our fine-tuning hyperparameters gives the best result, 0.937, comparing to all results of acceptor data, likewise, this architecture training with the first set of hyperparameter gives the best result, 0.902, of donor data. The NNs-based model results same trend as the 2nd CNNs-based model, however, it does not perform well on donor data with fine-tuning hyperparameters.

TABLE III. RESULTS FROM OUR PROPOSED MODELS OBTAINED FOR GWH DATASET

Type of RNA sequences	F1-Score					
	1st CNNs		2nd CNNs		NNs	
	Hyper-parameters #1	Hyper-parameters #1	Hyper-parameters #1	Hyper-parameters #2	Hyper-parameters #1	Hyper-parameters #2
Donor	0.688	**0.864**	0.902	0.646	0.901	0.192
Acceptor	0.872	**0.908**	0.687	**0.937**	0.814	0.901

TABLE IV. RESULTS OBTAINED FOR GWH DATASET COMPARING TO BENCHMARK RESULTS

Type of RNA sequences	F1-Score		
	Lee and Yoon's	SpliceRover's	Ours
Donor	0.816	0.907	0.864
Acceptor	0.753	0.873	**0.908**

To summarize, we illustrate our results comparing to other previous methods in Table 4. We show the results from the 1st CNNs-based model with our fine-tuning hyperparameters because it gives significantly better F1-score of both donor and acceptor data. We obtain the best F1-score, 0.908, for acceptor data. It is nearly 20% higher than Lee and Yoon's approach, and 4% better than the most recent approach as SpliceRover. In terms of donor data, our model performs better than Lee and Yoon's approach. Our result cannot compete with SpliceRover, though, we consider our proposed models show significantly improvement since we train models with less sample size than the testing data, GWH dataset, and different data preprocessing. They give similar results to benchmark approaches. Thus, there are rooms for potential improvement based on our proposed models further.

IV. CONCLUSIONS

In summary, we propose the new data preprocessing and improve deep-learning-based models for RNA splice sites recognition. Our data preprocessing trims down the long RNA sequences to the size of 40 nucleotides length with core-dinucleotides, GT and AG, in the middle. Our proposed models are based on CNNs and NNs. They are trained by using balanced dataset of RNA sequences. We perform experiments based on GWH dataset. The models significantly outperform previous methods, Lee and Yoon's and SpliceRover, by achieving F1-scores nearly 20% higher as maximum. In the future, we will try different data preprocessing methods, for example, shuffling the core-dinucleotides (GT and AG) to other positions, and shortening a sequence length. Another method is changing the way to transfer a sequence to a matrix, e.g. encoding two nucleotides instead of one. This, of course, requires more computational resources and running time. However, the matrix could lead to obtaining more abstractive level of the RNA sequence, perhaps, generating better results by expanding the subject's surrounding. We will also investigate pre-trained models (e.g. inception-v3 and mask R-CNN), and different architectures and hyperparameters, including evaluate on other RNA sequences data. Moreover, we will enhance data visualization methods to facilitate an interpretability of deep-learning-based models.

REFERENCES

[1] T. Lee and S. Yoon, "Boosted Categorical Restricted Boltzmann Machine for Computational Prediction of Splice Junctions," *Proc. 32nd Int. Conf. Mach. Learn.*, 2015.

[2] S. M. Berget, C. Moore, and P. A. Sharp, "Spliced segments at the 5′ terminus of adenovirus 2 late mRNA," *Proc. Natl. Acad. Sci.*, 1977.

[3] N. A. Faustino and T. A. Cooper, "Pre-mRNA splicing and human disease," *Genes and Development*. 2003.

[4] G. D. Stormo, "DNA binding sites: Representation and discovery," *Bioinformatics*. 2000.

[5] S. Degroeve, Y. Saeys, B. De Baets, P. Rouzé, and Y. Van de Peer, "SpliceMachine: Predicting splice sites from high-dimensional local context representations," *Bioinformatics*, 2005.

[6] S. Sonnenburg, G. Schweikert, P. Philips, J. Behr, and G. Rätsch, "Accurate splice site prediction using support vector machines," in *BMC Bioinformatics*, 2007.

[7] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nat. Biotechnol.*, 2015.

[8] J. Zhou and O. G. Troyanskaya, "Predicting effects of noncoding variants with deep learning–based sequence model," *Nat. Methods*, 2015.

[9] D. R. Kelley, J. Snoek, and J. L. Rinn, "Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks," *Genome Res.*, 2016.

[10] J. Zuallaert, F. Godin, M. Kim, A. Soete, Y. Saeys, and W. De Neve, "SpliceRover: interpretable convolutional neural networks for improved splice site prediction," *Bioinformatics*, 2018.

[11] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of Deep Learning and Reinforcement Learning to Biological Data," *IEEE Trans. Neural Networks Learn. Syst.*, 2018.

[12] Deeplearningandaiwinterschool.github.io. 2018. *Deep Learning and Artificial Intelligence Winter School 2018*. [online] Available at: https://deeplearningandaiwinterschool.github.io/dlai1.html.

[13] Deep Genomics. 2018. *Home | Deep Genomics*. [online] Available at: https://www.deepgenomics.com/.

[14] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[15] C. M. Bishop, Pattern Recognition and Machine Learning. 2006.