Chemometrics and Intelligent Laboratory Systems 188 (2019) 63–69

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

iSS-CNN: Identifying splicing sites using convolution neural network

Hilal Tayara [a], Muhammad Tahir [a,b,**], Kil To Chong [a,c,*]

[a] Department of Electronics and Information Engineering, Chonbuk National University, Jeonju 54896, South Korea
[b] Department of Computer Science, Abdul Wali Khan University, Mardan 23200, Pakistan
[c] Advanced Electronics and Information Research Center, Chonbuk National University, Jeonju 54896, South Korea

ARTICLE INFO

Keywords:
Computational biology
Deep learning
RNA
Splicing

ABSTRACT

RNA splicing is an important post-transcriptional modification of eukaryotic organisms in which a single gene can code for different proteins that have different biological functions. Thus, accurate identification of RNA splicing sites sequences is important for both drugs discovery and biomedical research. However, through laboratory techniques the discrimination of the splicing sites is very expensive. Therefore, an accurate computational model is needed. In this work, we introduce an efficient convolution neural network (CNN) model called iSS-CNN for splicing sites identification. Previous methods utilized hand-crafted features for identifying splicing sites, however, the proposed model extracts the features of the splicing sites automatically using the proposed CNN model. The performance of iSS-CNN has been evaluated on benchmark datasets and produced better outcomes than the existing methods. The iSS-CNN predictor obtained 96.66% of accuracy for a dataset containing splicing donor sites (SDS) and 93.57% of accuracy for a dataset containing splicing acceptor sites (SAS) using 5-fold cross-validation test. A webserver for the iSS-CNN tool has been established and made available at https://home.jbnu.ac.kr/NSCL/iss-cnn.htm.

1. Introduction

The transcription of pre-mRNA usually takes place from a eukaryotic gene template which consists of exons and one or more introns. In the pre-mRNA, the 5′ ends of introns are termed as splicing donor sites (SDS); whereas, the 3′ ends are called as splicing acceptor sites (SAS). The journey from pre-mRNA to mature RNA undergoes several biological processes as shown in Fig. 1. The exon part of final mRNA is the only region which is translated into proteins. Albeit, the removal of introns from both splicing sites i.e. 5′ and 3’ in pre-mRNA to form mRNA is important for both gene regulation and expression. Therefore, the identification of adequate splicing sites in pre-to mRNA processing is vitally important. The biochemical exploratory techniques can give some information related to the splicing sites, it is costly and time-consuming to depend on the biochemical exploratory approaches. Thus, it is a huge challenge to establish fast and precise computational models for identifying the splicing sites. In this circumstance, computational analysis splicing site tools have been developed such as iSS-Hyb-mRMR [1], SplicePredictor [2], NetGene [3,4], SplicePort [5], GeneSplicer [6], iSS-PseDNC [7], and iSS-PC [8]. In the last decades, according to a series of comprehensive reviews and several methods have been developed to identify splicing sites using machine learning methods. In these approaches, in 2001, Pertea et al., introduced a computational model for identification of splicing sites called GenSplicer [6]. The predictor has been effectively tested with two species, i.e. *Arabidopsis thaliana* and human DNA samples. The objective of this predictor was fusing already existing successful techniques such as maximal dependence decomposition (MDD) which was based on decision tree [9]. The MDD technique was enhanced by Markov model [10]. Further, it was considered that Markov model would also help to cover neighboring bases for dependencies near of splicing sites [6]. Zhang et al., introduced a sequence-based model for splicing sites detection using support vector machine as a classifier. In this regards, there were considerable evidence showing that the splicing site could be detected by interaction across exon instead of intron [11]. Similarly, Baten et al., presented a model for identification of splicing sites using probabilistic parameters and support vector machine. In this predictor, the Markov model was utilized and based on probabilistic parameters in the neighbors of splicing sites. These parameters were passed into support vector machine for splicing sites prediction [12]. Han et al., introduced a predicting model for predicting

* Corresponding author. Department of Electronics and Information Engineering, Chonbuk National University, Jeonju 54896, South Korea.
** Corresponding author. Department of Electronics and Information Engineering, Chonbuk National University, Jeonju 54896, South Korea.
E-mail addresses: mtahir@jbnu.ac.kr (M. Tahir), kitchong@jbnu.ac.kr (K.T. Chong).

https://doi.org/10.1016/j.chemolab.2019.03.002
Received 12 January 2019; Received in revised form 15 February 2019; Accepted 1 March 2019
Available online 19 March 2019
0169-7439/

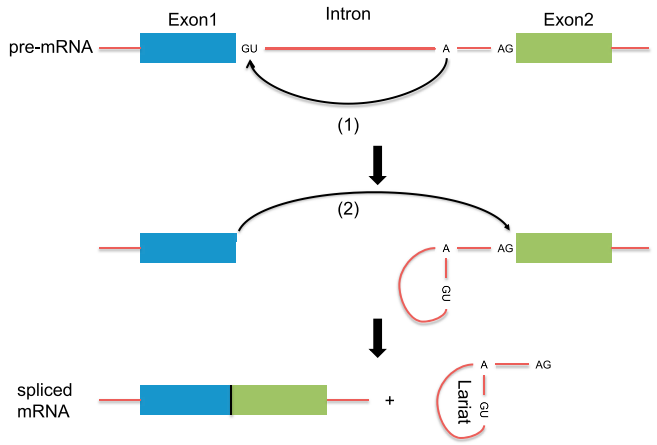

Fig. 1. Illustration of the two-step biochemistry of splicing.

splicing sites in the human genome. This model contained hybrid feature space of two different features extraction techniques namely increment of diversity (ID) along with position weight matrix (PWM). Then the resulted hybrid features were passed into support vector machine for classification of false and true splicing sites [13]. Chen et al., developed iSS-PseDNC predictor for prediction of splicing sites. In this model, a novel feature-vector was made by formulation called PseDNC in which six local structural DNA properties are passed to support vector machine for classification [7]. Iqbal et al., developed iSS-Hyb-mRMR predictor for identification of splicing sites, in this model the authors used two feature extraction techniques namely Tetranucleotide composition and Trinucleotide composition to extract numerical values from RNA samples. Different classification algorithms were applied such as PNN, KNN, GRNN and FitNet but the KNN classifier performed outstandingly. Then, they merged these two feature spaces to improve the discrimination power, after that mRMR was utilized to reduce the hybrid features spaces [1]. Most recently, Xu el at., developed iSS-PC computational model for prediction of splicing sites using twelve physical-chemical properties of PseDNC. This model achieved good results as compared to the exiting models [8]. Meanwhile, various computational methods and a powerful feature extraction technique called Type II PseKNC was applied in some DNA element identification and achieved good results [14,15]. But still these predictors have a large vacuum to accommodate further improvements like its prediction performance.

The above computational models and methods [16–26] need domain knowledge to hand-design the input features. As splicing site is affected by RNA samples, the system will automatically learn the features of splicing sites from these samples. This concept is achieved by utilization deep learning to extract the important features from multiple levels of abstraction. Deep learning has generated very successful results in natural language processing [27], speech recognition [28], information retrieval [29], and image recognition [30–32]. Currently, various RNA/DNA computational models have been introduced based on deep learning such as CNNclust [33], BiRen [34], DeepCpG [35], iDeepS [36], branch point selection [37], alternative splicing sites prediction [38], 2′-O-methylation sites prediction [39], etc.

In this study, a novel sequence-based splicing sites predictor is developed using the convolution neural networks (CNN). We propose an efficient architecture for splicing sites prediction called iSS-CNN. We search the best performing hyper-parameters using grid search method. The success rate of the iSS-CNN predictor is evaluated on two different benchmark datasets and outperforms the methods published recently in the literature.

According to Refs. [40–46], the researchers have mostly emphasized the guidelines of Chous 5-step rules that are: (i) benchmark datasets construction/selection; (ii) features extraction; (iii) operation engine; (iv) cross-validation test; (v) construct a web-server. Thus, these rules are followed in the study.

2. Materials and methods

In this section, we introduce the proposed model and benchmark datasets used for training and evaluation.

2.1. The proposed model

Previous works rely on handcrafted features extraction methods followed by a classifier for identifying splicing sites. On the other hand, the proposed model adapts convolutional neural networks (CNN) to identify splicing sites from RNA samples directly as CNN discovers the key features from the input RNA samples automatically during training. The proposed model is abbreviated as iSS-CNN and consists of different computational layers where the output of the last layer is used for prediction. Fig. 2 shows an illustration of iSS-CNN.

The input of iSS-CNN is a one-hot encoded vector of RNA sample, $s = \{s_1, .., s_n\}$ where $n = 140$ and $s_i \in \{A, T, C, G\}$, and the output is a real-valued score. The input vector has four channels {A, C, G, and T}, that are represented as (1 0 0 0), (0 1 0 0), (0 0 1 0), (0 0 0 1), respectively. Different hyper-parameters have been tuned during learning, the tuned hyper-parameters are: number of convolution layers, number of the filters, the size of the convolution filters, stride length, and the dropout probability. The ranges of these hyper-parameters are listed in Table 1. The best performing parameters have been selected based on the minimum validation loss. We select the top-4 best performing models. The configurations of the selected models are as follows:

- **Model-1**: It consists of a one-dimensional convolution layer with 16 filters with filter size of 7 and stride of 5, drop out layer with a

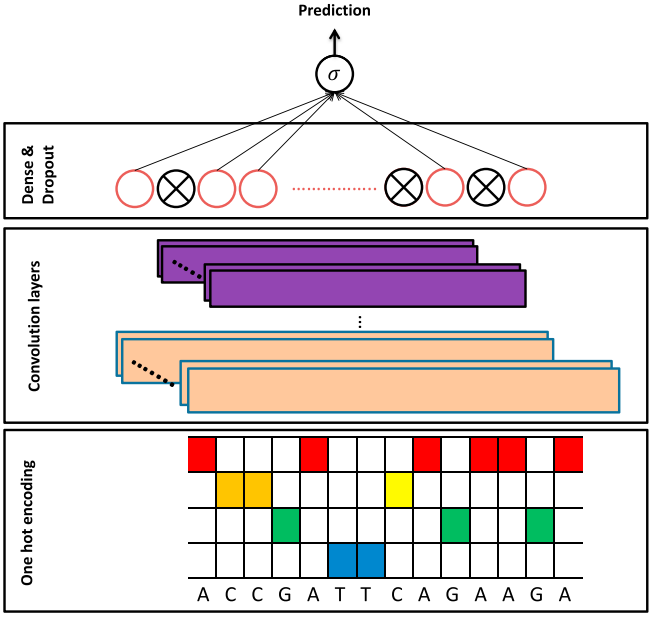

Fig. 2. Illustration of iSS-CNN model. It consists of convolution layers, dropout, and dense layer with a sigmoid activation function for prediction.

Table 1
Hyper-parameters tuning.

Hyper-Parameter	Range
Convolution layers	[1,2]
Filter size	[3,5,7,9]
Strides	[3],[3,5],[3,5,7],[3,5,7,9]
Number of filters	[8,16]
Dropout probability	[0.1,0.2,0.3]

probability of 0.2, and prediction layer which is a dense layer with sigmoid activation function.

- **Model-2**: It consists of a one-dimensional convolution layer with 16 filters with filter size of 7 and stride of 3, drop out layer with a probability of 0.3, and prediction layer which is a dense layer with sigmoid activation function.
- **Model-3**: It consists of two one-dimensional convolution layers with 16 filters with filter size of 9 and stride of 3 for both layers, drop out layer with a probability of 0.3, and prediction layer which is a dense layer with sigmoid activation function.
- **Model-4**: It consists of two one-dimensional convolution layers with 16 filters with filter size of 5 and stride of 3, drop out layer with a probability of 0.3, and prediction layer which is a dense layer with sigmoid activation function.

All convolution layers are followed by a nonlinear activation function called rectified linear unit (ReLU). The details of these models are given in Table 2.

In Table 2, the operator Conv1D(f, s, t) is a one-dimensional convolution operator where f is the number of the filters, s is the sizes of the filter, and t is the stride. Dropout(p) is a dropout operator with a probability of p. Dense(n) is a fully connected layer with n node. Sigmoid() function is a nonlinear activation function that squashes the output between zero and one.

The Convolutional layer is the basic unit that is used in our models. The first layer in the proposed models functions as a position weight matrix (PWM). Thus, convolution operation represents calculating the PWM scores with a sliding window with a step size equals to the stride on the sequence. More formally, convolution layer computes

$$conv(X)_{jk} = ReLU\left(\sum_{s=0}^{S-1}\sum_{f=0}^{F-1} W_{sf}^{k} X_{j+s,f}\right) \tag{1}$$

where X is the input, j is the index of the output position, and f is the index of the filter. Each convolution filter W^k is an $S \times F$ weight matrix where S represents the window size and F represents the number of input channels. For examples, F = 4 for the first convolution layer. ReLU denotes the rectified linear function and mathematically expressed as

$$ReLU(y) = \begin{cases} y & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases} \tag{2}$$

Table 2
The detailed architectures of the proposed models.

Model	Layer Type	Output shape
Model-1	Input	140 × 4
	Conv1D(16, 7, 5)	27 × 16
	Dropout(0.2)	432
	Dense(1)	1
	Sigmoid()	1
Model-2	Input	140 × 4
	Conv1D(16, 7, 3)	45 × 16
	Dropout(0.3)	720
	Dense(1)	1
	Sigmoid()	1
Model-3	Input	140 × 4
	Conv1D(16, 9, 3)	44 × 16
	Conv1D(16, 9, 3)	12 × 16
	Dropout(0.3)	192
	Dense(1)	1
	Sigmoid()	1
Model-4	Input	140 × 4
	Conv1D(16, 5, 3)	46 × 16
	Conv1D(16, 5, 3)	14 × 16
	Dropout(0.3)	224
	Dense(1)	1
	Sigmoid()	1

Desne layer with dropout: Dense layer transforms the feature vector z into a scalar output score. Let z be the incoming feature vector with $1 \times d$ dimension. Then, the output of the dense layer is

$$f = w_{d+1} + \sum_{k=1}^{d} w_k z_k \tag{3}$$

where w_{d+1} is an additive bias term, w_k is the weight of the z_k from the previous layer. Dropout is added to the dense layer which occasionally dropouts intermediate values from z_k by randomly assigning them to zero during training. Adding dropout during training time prevents the network from over-fitting and acts as a strong regularize. The Eq. (3) can be rewritten as

$$f = w_{d+1} + \sum_{k=1}^{d} m_k w_k z_k \text{ where } m_k \sim \text{ Bernoulli}(\alpha) \tag{4}$$

where Bernoulli(α) represents Bernoulli distribution with probability (α).

Sigmoid is the last layer and responsible for making predictions whether a given sequence is a splicing site or not as its output is scaled to the [0,1]. Sigmoid is expressed mathematically as:

$$Sigmoid(y) = \frac{1}{1 + e^{-y}} \tag{5}$$

The iSS-CNN is implemented by Keras framework [47]. The learning rate is set to 0.001. Adam optimizer is used for optimization. Weights are initialized using random uniform in the interval [-0.05, 0.05]. Batch size is set to 16. Number of Epochs is set to 200 and early stopping is applied based on validation loss.

2.2. Benchmark dataset

In this study, the dataset contains human splicing sites sequences and consists of two parts: the first part consists of splicing donor site (SDS) sequences while the second part contains splicing acceptor site (SAS) sequences. It can be numerically expressed as below:

$$S_1 = S_1^+ \cup S_1^- \tag{6}$$

$$S_2 = S_2^+ \cup S_2^- \tag{7}$$

Where S_1 represents a splicing donor site (SDS) and S_2 represents a splicing acceptor site (SAS). The positive dataset S_1^+ consists of 2,796 true SDS sequences while the negative dataset S_1^- consists of 2,800 false SDS sequences; similarly S_2^+ contains 2,880 true SAS sequences, while S_2^- contains 2,800 false SAS sequences. The symbol ∪ denotes the union in the set theory. These datasets can be obtained from Ref. [7].

2.3. Cross-validation

In statistical prediction, the error rate is used to evaluate the performance of classification algorithms. For this purpose the cross-validation technique is followed, where the whole dataset is partitioned into mutually exclusive folds [15,48–54]. In the k-fold cross-validation test, a particular dataset can be divided into k-fold for cross-validation. One fold is reserved for the testing purpose, while the k-1 folds are used for training the proposed model. This is a k-times recursive process, where for testing every fold is used at least once. Five-fold, seven-fold and ten-fold subsampling test are usually used. Therefore, we applied a 5-fold cross-validation test to measure the performance of the iSS-CNN. For S_1 dataset, we randomly distribute the dataset and into five equal size subset, mathematically expressed below as:

$$S_1^+ = S_{11}^+ \cup S_{12}^+ \cup S_{13}^+ \cup S_{14}^+ \cup S_{15}^+ \tag{8}$$

$$S_1^- = S_{11}^- \cup S_{12}^- \cup S_{13}^- \cup S_{14}^- \cup S_{15}^- \quad (9)$$

where S_{1i}^+ is a subset of S_1^+ for $i = 1, 2, 3, 4, 5$, and S_{1i}^- is a subset of S_1^- for $i = 1, 2, 3, 4, 5$.

In addition, the number of the elements in S_{1i}^+ and S_{1i}^- satisfies the following condition:

$$|S_{11}^+| \approx |S_{12}^+| \approx |S_{13}^+| \approx |S_{14}^+| \approx |S_{15}^+| \quad (10)$$

$$|S_{11}^-| \approx |S_{12}^-| \approx |S_{13}^-| \approx |S_{14}^-| \approx |S_{15}^-| \quad (11)$$

where $|S_{1i}^-|$ is the number of the elements in the subset S_{1i}^- for $i = 1, 2, 3, 4, 5$. Five subsets are obtained from the dataset S_1 as shown below

$$S_1 = S_1^{'} \cup S_2^{'} \cup S_3^{'} \cup S_4^{'} \cup S_5^{'} \quad (12)$$

where $S_i^{'} = S_{1i}^+ \cup S_{1i}^-$ for $i = 1, 2, 3, 4, 5$ with

$$|S_1^{'}| \approx |S_2^{'}| \approx |S_3^{'}| \approx |S_4^{'}| \approx |S_5^{'}| \quad (13)$$

Thus, one set from Eq. (12) is chosen for test and the remaining for training. Cross-validation is repeated five times and the average scores among the outputs are considered as the final outcome. In the same manner, cross-validation is applied for the SAS dataset S_2 of Eq. (7).

Table 3
The results of SDS sequences based on different models' configurations.

Model	ACC(%)	MCC(%)	Sn(%)	Sp(%)
Model-1	95.14	90.30	95.75	94.48
Model-2	95.59	91.17	95.65	95.52
Model-3	**96.66**	**93.32**	**97.23**	**96.06**
Model-4	96.16	92.34	96.82	95.47

Bold format represents the highest obtained result.

Table 4
The results of SAS sequences based on different models' configurations.

Model	ACC(%)	MCC(%)	Sn(%)	Sp(%)
Model-1	91.94	83.96	93.95	89.86
Model-2	**93.57**	**87.19**	**95.16**	**91.94**
Model-3	92.76	85.64	94.64	90.84
Model-4	92.99	86.00	93.46	92.52

Bold format represents the highest obtained result.

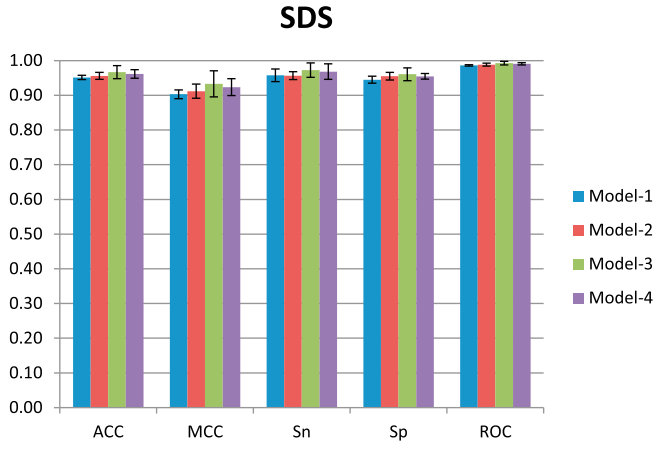

Fig. 3. The test results achieved from different models on benchmark dataset only containing SDS sequences.

3. Results and discussion

3.1. Evaluation metrics

The following four metrics are widely utilized to check the performance of the prediction model [17,55–57] namely: Matthew correlation coefficient (MCC), accuracy (ACC), specificity (Sp), and sensitivity (Sn):

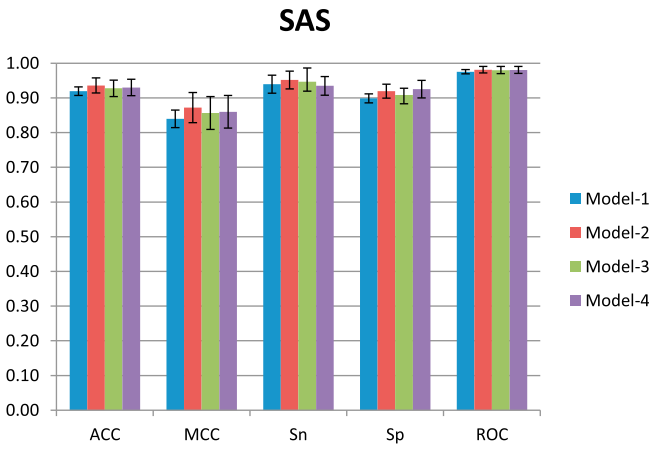

Fig. 4. The test results achieved from different models on benchmark dataset only containing SAS sequences.

Table 5
The test results on SDS using different methods.

Method	ACC(%)	MCC(%)	Sn(%)	Sp(%)	ROC-AUC(%)
iSS-PseDNC	87.71	75.46	89.56	85.86	92.39
iSS-PC	90.56	81.56	90.09	91.04	95.66
iSS-CNN	**96.66**	**93.32**	**97.23**	**96.06**	**99.26**

Bold format represents the highest obtained result.

Table 6
The test results on SAS using different methods.

Method	ACC(%)	MCC(%)	Sn(%)	Sp(%)	ROC-AUC(%)
iSS-PseDNC	88.73	77.89	94.24	83.07	95.18
iSS-PC	91.11	82.24	90.14	**92.11**	96.28
iSS-CNN	**93.57**	**87.19**	**95.16**	91.94	**98.11**

Bold format represents the highest obtained result.

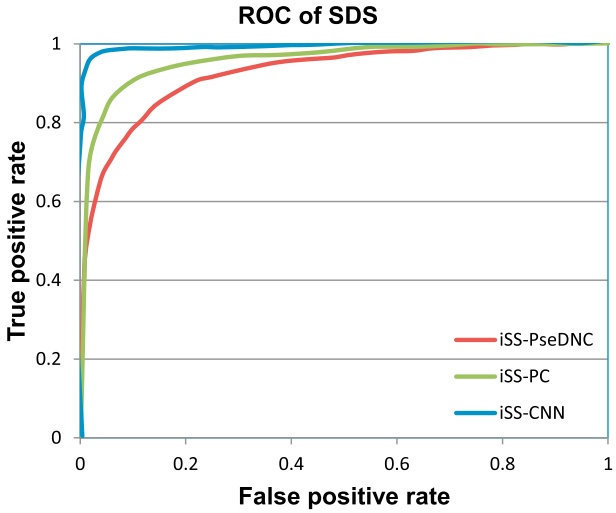

Fig. 5. ROC curves of the SDS sequences using different predictors.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (14)$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

$$Sp = \frac{TN}{TN + FP} \quad (16)$$

$$Sn = \frac{TP}{TP + FN} \quad (17)$$

Where; TP: True positive which represents correctly identified splicing sites, TN: True negative which represents correctly rejected splicing sites, FP: False positive which represents incorrectly identified splicing sites, FP: False negative which represents incorrectly rejected splicing sites.

In addition, receiver operating characteristic (ROC) curves, a graphical form for visualizing the performance of the proposed models, is used. The larger the AUC the better model's performance [58,59].

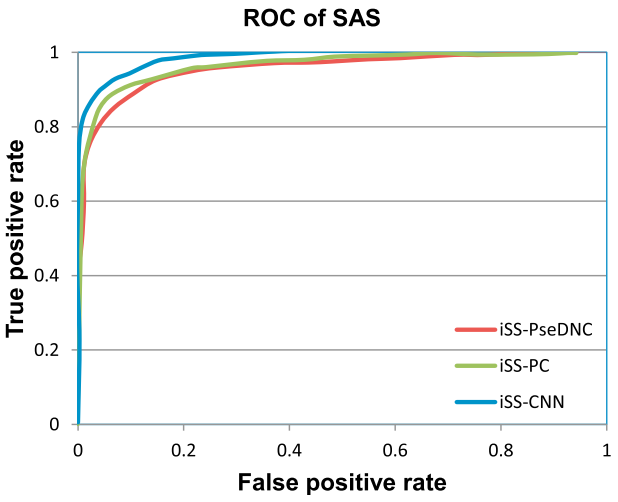

Fig. 6. ROC curves of the SAS sequences using different predictors.

3.2. Results and discussion

As introduced in Section 2.1, we select the top-4 best performing models resulted from hyper-parameters search. Model-1 and Model-2 contain one convolution layer. On the other hand, Model-3 and Model-4 contain two convolution layers with different configurations.

Table 3 shows the performance results of the proposed models of SDS and Table 4 shows the performance results of the proposed models of SAS. We can see that the best performing model on SDS benchmark is Model-3. On the other hand, Model-2 is the best performing model on SAS benchmark.

The 5-fold cross-validation test results for different models are shown in Fig. 3 for SDS and Fig. 4 for SAS with standard deviation.

In addition, we compare our proposed model with iSS-PseDNC [7] and iSS-PC [8]. Table 5 shows the comparison of 5-fold cross-validation test results with the aforementioned systems on benchmark dataset only containing SDS sequences. The results show that iSS-CNN predictor outperforms the existing predictors in terms of all comparison metrics. More specifically, we improve the accuracy (ACC) by 6.1%, Matthew correlation coefficient (MCC) by 11.76%, sensitivity (Sn) by 7.14%,

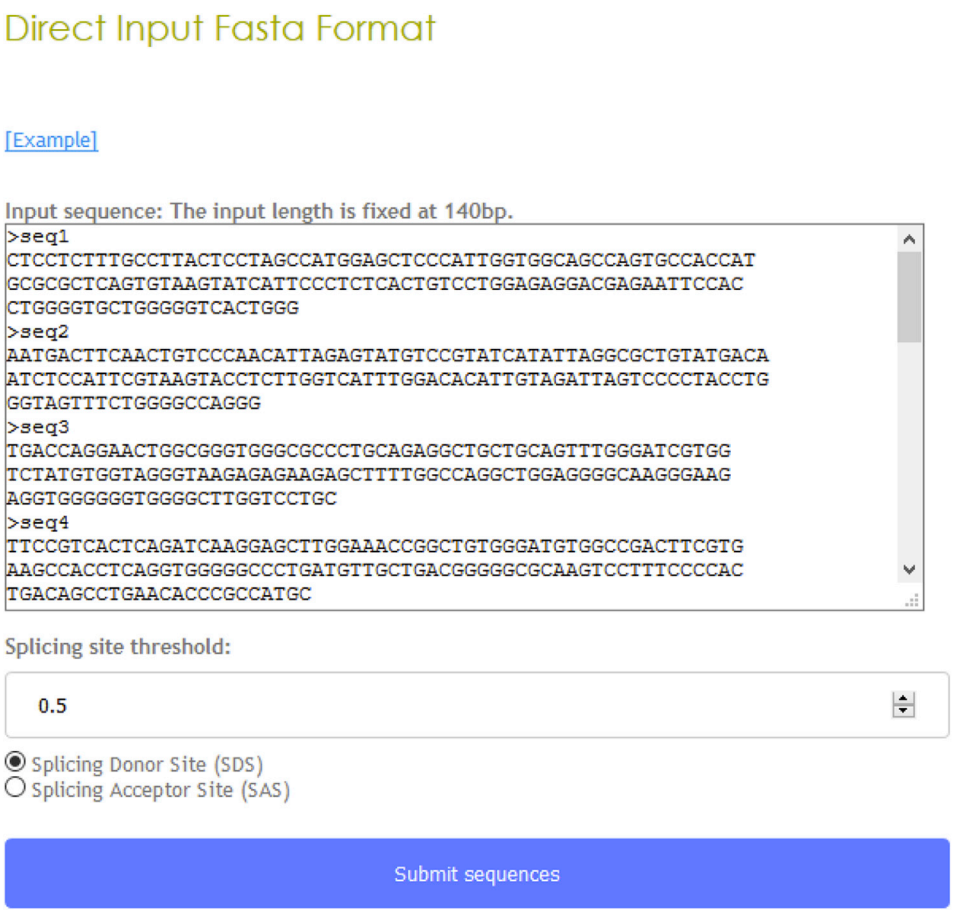

Fig. 7. Webserver homepage: Direct sequence input in fasta format.

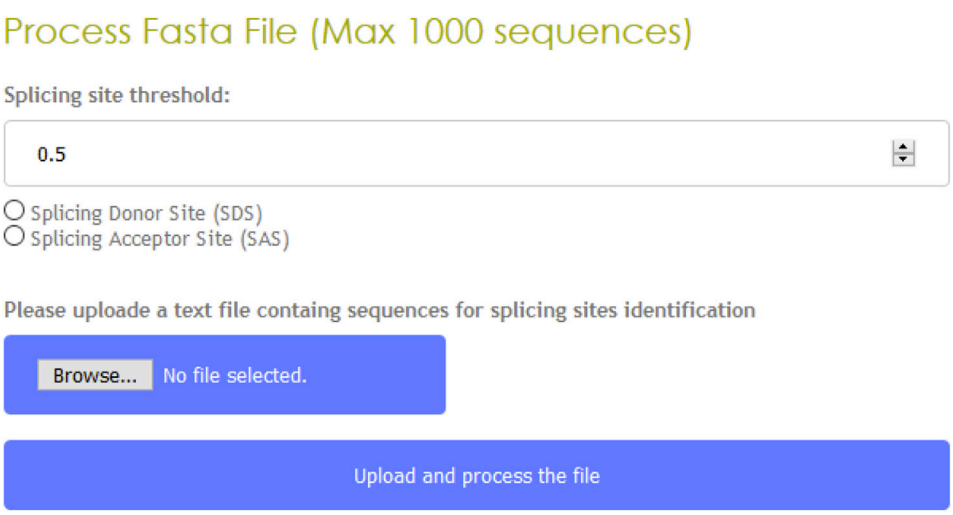

Fig. 8. Webserver homepage: Uploading and processing a file containing splicing sequences.

specificity (Sp) by 5.02%, and ROC-AUC by 3.6%.

Also, Table 6 shows the comparison of 5-fold cross-validation test results with the aforementioned systems on benchmark dataset only containing SAS sequences. It can be also observed that the iSS-CNN outperforms the other predictors. More specifically, we improve the accuracy (ACC) by 2.46%, Matthew correlation coefficient (MCC) by 4.95%, sensitivity (Sn) by 0.92%, and ROC AUC by 1.83%.

The ROC-AUC curves for the proposed models, iSS-PseDNC [7] and iSS-PC [8] are shown in Fig. 5 and Fig. 6 for SDS and SAS, respectively. It can also be seen that our proposed models achieve better ROC-AUC compared to the state-of-the-art models.

3.3. Webserver

We have established a webserver for the proposed method "iSS-CNN" as shown in Figs. 7 and 8 and made it available at (https://home.jbnu.ac.kr/NSCL/iss-cnn.htm). The webserver is user-friendly and easy to use. It supports obtaining the prediction by direct input of the sequences as shown in Fig. 7 and by uploading the sequences in FASTA format as shown in Fig. 8. In both cases, the user has to select the splice site type such as splice donor site (SDS) or splice acceptor site (SAS). In addition, we give an option to decide the cut-off threshold where the default value is set to 0.5.

4. Conclusion

RNA splicing is an important biological process that comprises interactions among DNA, RNA, and proteins. Therefore, an efficient and accurate computational model namely iSS-CNN was developed to identify splicing sites. It is based on a convolution neural network. The iSS-CNN extracts the features automatically from raw DNA/RNA sequences. The hyper-parameter searching has been followed in order to figure out the most optimal models. The performance shows that iSS-CNN is more stable and accurate than the other comparative methods in terms of all evaluation parameters. It is expected that iSS-CNN might be useful in drug-related applications and academia. Finally, we have established a webserver for the proposed iSS-CNN model at (https://home.jbnu.ac.kr/NSCL/iss-cnn.htm).

Conflicts of interest

The authors declare no conflict of interest.

Funding

This research was supported by the Brain Research Program of the National Research Foundation (NRF) funded by the Korean government (MSIT) (No. NRF-2017M3C7A1044815).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2019.03.002.

References

[1] M. Iqbal, M. Hayat, iss-hyb-mrmr: identification of splicing sites using hybrid space of pseudo trinucleotide and pseudo tetranucleotide composition, Comput. Methods Progr. Biomed. 128 (2016) 1–11.
[2] V. Brendel, J. Kleffe, Prediction of locally optimal splice sites in plant pre-mrna with applications to gene identification in arabidopsis thaliana genomic dna, Nucleic Acids Res. 26 (20) (1998) 4748–4757.
[3] S.M. Hebsgaard, P.G. Korning, N. Tolstrup, J. Engelbrecht, P. Rouzé, S. Brunak, Splice site prediction in arabidopsis thaliana pre-mrna by combining local and global sequence information, Nucleic Acids Res. 24 (17) (1996) 3439–3452.
[4] S. Brunak, J. Engelbrecht, S. Knudsen, Prediction of human mrna donor and acceptor sites from the dna sequence, J. Mol. Biol. 220 (1) (1991) 49–65.
[5] R.I. Dogan, L. Getoor, W.J. Wilbur, S.M. Mount, Spliceportan interactive splice-site analysis tool, Nucleic Acids Res. 35 (suppl_2) (2007) W285–W291.
[6] M. Pertea, X. Lin, S.L. Salzberg, Genesplicer: a new computational method for splice site prediction, Nucleic Acids Res. 29 (5) (2001) 1185–1190.
[7] W. Chen, P.-M. Feng, H. Lin, K.-C. Chou, iss-psednc: identifying splicing sites using pseudo dinucleotide composition, BioMed Res. Int. (2014), 623149.
[8] Z.-C. Xu, P. Wang, W.-R. Qiu, X. Xiao, iss-pc: identifying splicing sites via physical-chemical properties using deep sparse auto-encoder, Sci. Rep. 7 (1) (2017) 8222.
[9] C. Burge, S. Karlin, Prediction of complete gene structures in human genomic dna1, J. Mol. Biol. 268 (1) (1997) 78–94.
[10] S. Salzberg, A.L. Delcher, K.H. Fasman, J. Henderson, A decision tree system for finding genes in dna, J. Comput. Biol. 5 (4) (1998) 667–680.
[11] X.H. Zhang, K.A. Heller, I. Hefter, C.S. Leslie, L.A. Chasin, Sequence information for the splicing of human pre-mrna identified by support vector machine classification, Genome Res. 13 (12) (2003) 2637–2650.
[12] A.K. Baten, B.C. Chang, S.K. Halgamuge, J. Li, Splice site identification using probabilistic parameters and svm classification, BMC Bioinf. 7 (2006) S15. BioMed Central.
[13] J. Han, Y. Cui, J. Liu, X. Zhang, An effective computational method for human splice sites identification, in: Control Conference (ASCC), 2013 9th Asian, IEEE, 2013, pp. 1–4.
[14] F.-Y. Dao, H. Lv, F. Wang, C.-Q. Feng, H. Ding, W. Chen, H. Lin, Identify origin of replication in saccharomyces cerevisiae using two-step feature selection technique, Bioinformatics (2018) bty943.
[15] C.-Q. Feng, Z.-Y. Zhang, X.-J. Zhu, Y. Lin, W. Chen, H. Tang, H. Lin, iterm-pseknc: a sequence-based tool for predicting bacterial transcriptional terminators, Bioinformatics (2018) bty827.
[16] M. Tahir, M. Hayat, S.A. Khan, A two-layer computational model for discrimination of enhancer and their types using hybrid features pace of pseudo k-tuple nucleotide composition, Arabian J. Sci. Eng. (2017) 1–9.
[17] S.-H. Guo, E.-Z. Deng, L.-Q. Xu, H. Ding, H. Lin, W. Chen, K.-C. Chou, inuc-pseknc: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition, Bioinformatics 30 (11) (2014) 1522–1529.
[18] M. Kabir, M. Hayat, irspot-gaensc: identifing recombination spots via ensemble classifier and extending the concept of chous pseaac to formulate dna samples, Mol. Genet. Genom. 291 (1) (2016) 285–296.

[19] M. Hayat, A. Khan, Predicting membrane protein types by fusing composite protein sequence features into pseudo amino acid composition, J. Theor. Biol. 271 (1) (2011) 10–17.
[20] M. Tahir, M. Hayat, M. Kabir, Sequence based predictor for discrimination of enhancer and their types by applying general form of chou's trinucleotide composition, Comput. Methods Progr. Biomed. 146 (2017) 69–75.
[21] Z. Liu, X. Xiao, W.-R. Qiu, K.-C. Chou, idna-methyl: identifying dna methylation sites via pseudo trinucleotide composition, Anal. Biochem. 474 (2015) 69–77.
[22] B. Liu, L. Fang, R. Long, X. Lan, K.-C. Chou, ienhancer-2l: a two-layer predictor for identifying enhancers and their strength by pseudo k-tuple nucleotide composition, Bioinformatics 32 (3) (2015) 362–369.
[23] Y.-N. Fan, X. Xiao, J.-L. Min, K.-C. Chou, inr-drug: Predicting the interaction of drugs with nuclear receptors in cellular networking, Int. J. Mol. Sci. 15 (3) (2014) 4915–4937.
[24] W.-C. Li, E.-Z. Deng, H. Ding, W. Chen, H. Lin, iori-pseknc: a predictor for identifying origin of replication with pseudo k-tuple nucleotide composition, Chemometr. Intell. Lab. Syst. 141 (2015) 100–106.
[25] X. Xiao, H.-X. Ye, Z. Liu, J.-H. Jia, K.-C. Chou, iros-gpseknc: predicting replication origin sites in dna by incorporating dinucleotide position-specific propensity into general pseudo nucleotide composition, Oncotarget 7 (23) (2016) 34180.
[26] B. Liu, S. Wang, R. Long, K.-C. Chou, irspot-el: identify recombination spots with an ensemble learning approach, Bioinformatics 33 (1) (2016) 35–41.
[27] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, J. Mach. Learn. Res. 12 (Aug) (2011) 2493–2537.
[28] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[29] W. Qu, D. Wang, S. Feng, Y. Zhang, G. Yu, A novel cross-modal hashing algorithm based on multimodal deep learning, Sci. China Inf. Sci. 60 (9) (2017), 092104.
[30] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[31] H. Tayara, K.G. Soo, K.T. Chong, Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network, IEEE Access 6 (2018) 2220–2230.
[32] H. Tayara, K. Chong, Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network, Sensors 18 (10) (2018) 3341.
[33] G. Aoki, Y. Sakakibara, Convolutional neural networks for classification of alignments of non-coding rna sequences, Bioinformatics 34 (13) (2018) i237–i244, https://doi.org/10.1093/bioinformatics/bty228.
[34] B. Yang, F. Liu, C. Ren, Z. Ouyang, Z. Xie, X. Bo, W. Shu, Biren: predicting enhancers with a deep-learning-based model using the dna sequence alone, Bioinformatics 33 (13) (2017) 1930–1936, https://doi.org/10.1093/bioinformatics/btx105.
[35] C. Angermueller, H.J. Lee, W. Reik, O. Stegle, Deepcpg: accurate prediction of single-cell dna methylation states using deep learning, Genome Biol. 18 (1) (2017) 67.
[36] X. Pan, P. Rijnbeek, J. Yan, H.-B. Shen, Prediction of rna-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks, BMC Genomics 19 (1) (2018) 511.
[37] I. Nazari, H. Tayara, K.T. Chong, Branch point selection in rna splicing using deep learning, IEEE Access 7 (2019) 1800–1807, https://doi.org/10.1109/ACCESS.2018.2886569.
[38] M. Oubounyt, Z. Louadi, H. Tayara, K.T. Chong, Deep learning models based on distributed feature representations for alternative splicing prediction, IEEE Access 6 (2018) 58826–58834, https://doi.org/10.1109/ACCESS.2018.2874208.
[39] M. Tahir, H. Tayara, K.T. Chong, irna-pseknc(2methyl): identify rna 2'-o-methylation sites by convolution neural network and chou's pseudo components, J. Theor. Biol. 465 (2019) 1–6. https://doi.org/10.1016/j.jtbi.2018.12.034. http://www.sciencedirect.com/science/article/pii/S0022519318306349.
[40] L. Cai, T. Huang, J. Su, X. Zhang, W. Chen, F. Zhang, L. He, K.-C. Chou, Implications of newly identified brain eqtl genes and their interactors in schizophrenia, Mol. Ther. Nucleic Acids 12 (2018) 433–442.
[41] J. Song, Y. Wang, F. Li, T. Akutsu, N.D. Rawlings, G.I. Webb, K.-C. Chou, iprot-sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites, Briefings Bioinf. (2018) bby028.
[42] Y. Zhang, R. Xie, J. Wang, A. Leier, T.T. Marquez-Lago, T. Akutsu, G.I. Webb, K.-C. Chou, J. Song, Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework, Briefings Bioinf. 5 (2018).
[43] W. Chen, P. Feng, H. Ding, H. Lin, K.-C. Chou, irna-methyl: identifying n6-methyladenosine sites using pseudo nucleotide composition, Anal. Biochem. 490 (2015) 26–33.
[44] K.-C. Chou, H.-B. Shen, Recent advances in developing web-servers for predicting protein attributes, Nat. Sci. 1 (02) (2009) 63.
[45] K.-C. Chou, Impacts of bioinformatics to medicinal chemistry, Med. Chem. 11 (3) (2015) 218–234.
[46] K.-C. Chou, An unprecedented revolution in medicinal chemistry driven by the progress of biological science, Curr. Top. Med. Chem. 17 (21) (2017) 2337–2358.
[47] F. Chollet, et al., Keras, 2015. https://keras.io.
[48] M. Tahir, M. Hayat, Machine learning based identification of protein–protein interactions using derived features of physiochemical properties and evolutionary profiles, Artif. Intell. Med. 78 (2017) 61–71.
[49] M. Tahir, M. Hayat, S.A. Khan, inuc-ext-psetnc: an efficient ensemble model for identification of nucleosome positioning by extending the concept of chous pseaac to pseudo-tri-nucleotide composition, Mol. Genet. Genom. 294 (1) (2019) 199–210.
[50] M. Hayat, A. Khan, Memhyb: predicting membrane protein types by hybridizing saac and pssm, J. Theor. Biol. 292 (2012) 93–102.
[51] M. Hayat, A. Khan, M. Yeasin, Prediction of membrane proteins using split amino acid and ensemble classification, Amino Acids 42 (6) (2012) 2447–2460.
[52] M. Hayat, M. Tahir, Psofuzzysvm-tmh: identification of transmembrane helix segments using ensemble feature space by incorporated fuzzy support vector machine, Mol. Biosyst. 11 (8) (2015) 2255–2262.
[53] M. Waris, K. Ahmad, M. Kabir, M. Hayat, Identification of dna binding proteins using evolutionary profiles position specific scoring matrix, Neurocomputing 199 (2016) 154–162.
[54] M. Kabir, M. Iqbal, S. Ahmad, M. Hayat, itis-pseknc: identification of translation initiation site in human genes using pseudo k-tuple nucleotides composition, Comput. Biol. Med. 66 (2015) 252–257.
[55] H. Lin, E.-Z. Deng, H. Ding, W. Chen, K.-C. Chou, ipro54-pseknc: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition, Nucleic Acids Res. 42 (21) (2014) 12961–12972.
[56] W. Chen, P.-M. Feng, E.-Z. Deng, H. Lin, K.-C. Chou, itis-psetnc: a sequence-based predictor for identifying translation initiation site in human genes using pseudo trinucleotide composition, Anal. Biochem. 462 (2014) 76–83.
[57] B. Liu, F. Liu, X. Wang, J. Chen, L. Fang, K.-C. Chou, Pse-in-one: a web server for generating various modes of pseudo components of dna, rna, and protein sequences, Nucleic Acids Res. 43 (W1) (2015) W65–W71.
[58] J. Grau, I. Grosse, J. Keilwagen, Prroc: computing and visualizing precision-recall and receiver operating characteristic curves in r, Bioinformatics 31 (15) (2015) 2595–2597.
[59] T. Fawcett, Roc graphs: notes and practical considerations for researchers, Mach. Learn. 31 (1) (2004) 1–38.