IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 19, 2018, accepted October 1, 2018, date of publication October 8, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2874208

Deep Learning Models Based on Distributed Feature Representations for Alternative Splicing Prediction

MHANED OUBOUNYT[1,*], ZAKARIA LOUADI[1,*], HILAL TAYARA[1], AND KIL TO CHONG[2]

[1]Department of Information and Electronics Engineering, Chonbuk National University, Jeonju 54896, South Korea
[2]Division of Electronic Engineering, and Advanced Research Center of Electronics and Information, Chonbuk National University, Jeonju 54896, South Korea

Corresponding authors: Hilal Tayara (hilaltayara@jbnu.ac.kr) and Kil To Chong (kitchong@jbnu.ac.kr)

This work was supported by the Brain Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT, & Future Planning under Grant NRF-2017M3C7A1044815.

*These authors contributed equally to this work.

ABSTRACT Alternative splicing (AS) is a fundamental step in mRNA maturation and gene expression. The advancement in RNA sequencing technologies has shed light on the role of AS in increasing protein isoform diversity. AS is recognized to be involved in the regulation of both physiological and pathological functions, hence it is an essential part of the study of gene regulation development and diseases. With the recent advances in machine learning, there is an interest in developing accurate deep learning based computational models for AS prediction. In this paper, we propose a convolutional neural network and multilayer perceptron models to tackle the AS prediction task as classification and regression. These models use feature representations learned from genomic data and cellular context. Unlike previous works which use hand-crafted feature extraction, we propose an automatic feature learning approach to avoid explicit and predefined feature extraction. The proposed approach is based on the adaptation of two extensively used natural language processing techniques, namely word2vec and doc2vec. In order to understand the effects of different representation learning techniques, many experiments have been conducted to predict AS based on the cassette exons and cell type. Overall, experimental results on five tissues data set prove that learning features from genome sequence add a significant improvement to AS outcome prediction in both classification and regression tasks.

INDEX TERMS Alternative splicing (AS), convolution neural network (CNN), cassette exons, feature representations.

I. INTRODUCTION

Alternative splicing (AS) is a major regulated mechanism during gene expression process, where the exons of a primary transcript are spliced together while the introns are cut out. During splicing an alternative exon may be included or excluded, allowing a single gene to give rise to numerous splicing isoforms. Consequently, different proteins can be produced which in turn contribute to the enrichment of cellular protein diversity [1]. For example, AS permits the human genome to produce more than 100,000 different proteins from only 20,000 protein-coding genes [2]. Scientists observed seven main types of AS, of which the most common type is exon skipping (ES) [3]. Splicing event is categorized as ES type when the processed pre-mRNA sequence is a cassette exons - Cassette exons consists of an alternative exon flanked by two constitutive exons - and the resulting mRNA fragment can either skip or comprise the alternative exon. AS derives its huge importance form the fact that approximately 90 % to 95 % of human multi-exon genes are alternatively spliced [2], [4].

Splicing pre-mRNA into mRNA is a specific process that results in a particular protein essential for the cell function. Any error in this process can defect the encoded protein and thus AS is highly linked to diseases and therapy. An increasing number of experiments illustrates that any AS miss-regulation or disruption of the regular process of splicing can cause a broad range of diseases [5], [6], such as cancer. The rising knowledge about splicing regulation allows the development of effective treatment options.

The existence or absence of some regulatory elements in the cis-regulatory region alongside with splicing factors can affect the inclusion rate of an alternative exon in the final mature mRNA [7]. Take all these factors into consideration would be beneficial for predicting the percentage of splicing inclusion (PSI, Ψ) for a given alternative exon. Ψ helps in discriminating the isoforms that include the alternative exon and those that exclude it.

In the recent years, there is a growing interest among researchers in establishing methods for predicting AS. However, the existing overlap among AS mechanism and other mechanisms limits the ability to analyze AS and cause an incomplete understanding of its regulation [3]. On the other hand, there is a scientific consensus, based on many experimental observations such as the one conducted in [8], that AS is a non-random process. Moreover, ''splicing code'' is proposed as features set extracted from the genome and RNA characteristics [9]. These features reveal splicing patterns of any given primary transcript of a particular cell.

Previous studies introduced computational models that address the issue of predicting AS. Bayesian neural network (BNN) was used by Xiong *et al.* [10]. They have used 3665 cassette exons from which they have extracted 1014 RNA features. Using BNN helped in avoiding overfitting problem with the comparison to other traditional machine learning algorithms such as support vector machines (SVM), multinomial logistic regression (MLR), K-nearest neighbors, and naive Bayes.

Leung *et al.* [11] proposed a computational model based on deep neural network (DNN). They have extracted 1393 features from the exons and adjacent introns of each input sequence. The model was capable of dealing with complex relationships exists in the biological dataset. The reported performance of DNN outperformed BNN.

Recently, the proposed work in [12] introduced more accurate models based on deep neural network (DNN). Also, they have reconstructed the BNN model proposed in [11] to establish a baseline. These models achieved better accuracy in PSI prediction due to the following reasons i) the newly designed target function improves the performance because it allows directly predicting PSI instead of using categories ii) extension of the data to 27 mouse tissues compared to 5 tissues used in [11].

A range of studies attempted to find the perfect feature extraction and representation techniques to predict splicing patterns. Barash *et al.* [9] developed the ''splicing code'' which extracts 1014 RNA features from any alternative exon of interest and its surrounding introns and exons. This method was extended later by Barash *et al.* [13], and the AVISPA tool has been introduced. This tool extracts features from any given exon and its nearby sequence and directly indicates whether the exon is alternatively spliced or not. This approach of features extraction has since been adopted by other works including [12].

The ultimate objective is to estimate the alternative exon inclusion level in given alternative cassette exons indicated by the level of percent spliced-in PSI or Ψ (i.e., in ES case it is the inclusion percentage of the alternative exon). In this work, we have revealed that the AS dilemma mostly lies in the feature extraction method. A large number of features is not necessarily beneficial for the model instead it may include many noisy irrelevant features. Extraction of an optimal feature set can lead to better results. Therefore, we have adopted an automatic features learning method based on word embedding. Two frameworks namely word2vec and doc2vec were used to learn these embedding representations. Different architectures were explored to find out the perfect features representation. We perform the evaluation of different deep learning models on a dataset of 5 tissues for both regression and classification tasks.

The main contribution of this work was to propose two variant models for feature representation of AS. This has resulted in achieving better AS prediction accuracy compared to other methods.

The remainder of this paper is organized as follows: Section II provides data processing, proposed models, and implementation details. Section III presents experimental results, evaluations procedure, comparison, and discussions. The paper is concluded in Section IV.

II. METHODOLOGY

The pipeline of our approach consists of three main steps: data processing that decomposes each biological sequence to a sentence of words after quantification Fig. 2(a), (b), the feature representation stage that maps each sentence to a feature set Fig. 2(c), and deep learning based computational models that predict alternative splicing Fig. 2(d). The whole process is illustrated in Fig. 2 and each step is described in details respectively in this section.

A. DATA PREPROCESSING

The dataset used in this work consists of approximately 11,000 mouse cassette exons for each one of the five tissues. This data is taken from the RNA-seq data provided by Brawand *et al.* [14]. The five tissues are as follow: heart, brain, kidney, liver, and testis. We start by aligning the RNA-Seq reads to the genome using STAR aligner [15]. Next, We perform PSI quantification using MAJIQ tool [16], which is a probabilistic model that takes as input RNA-seq and the transcription annotation files and provide quantification and visualization of the local splicing variations (LSV). MAJIQ allows generation of Ψ value for each cassette exons and removes duplicate cassettes from the dataset. It was also used for the same purpose in [12]. In order to use the same target function as in [11], the data was divided into 3 classes based on the PSI value: low ($0 \leq \Psi < 0.33$), medium ($0.33 \leq \Psi < 0.66$), and high ($0.66 \leq \Psi \leq 1$).

Several works, including [9], have investigated the distribution of splicing features across the sequences and reported that most AS distinctive features are frequently located within 300 nucleotides (nt) of upstream and/or downstream introns of the concerned exons. Accordingly, we process each

cassette exons in a way that conserves the 3 exons CE1, AE, and CE2. If introns are longer than 600 nt we include 300 nt from the CE1 downstream intron sequence, 300 nt from each of upstream and downstream introns of AE, and 300 nt from the upstream intron of CE2. Otherwise, the whole cassette exon sequence is included. These parts are denoted P1, P2, P3, and P4 respectively in Fig. 1(a). Next, every cassette exon is represented by a fixed dimension feature vector using a distributed representation model. As shown in Fig. 1(b). This process is described in detail in the next section.

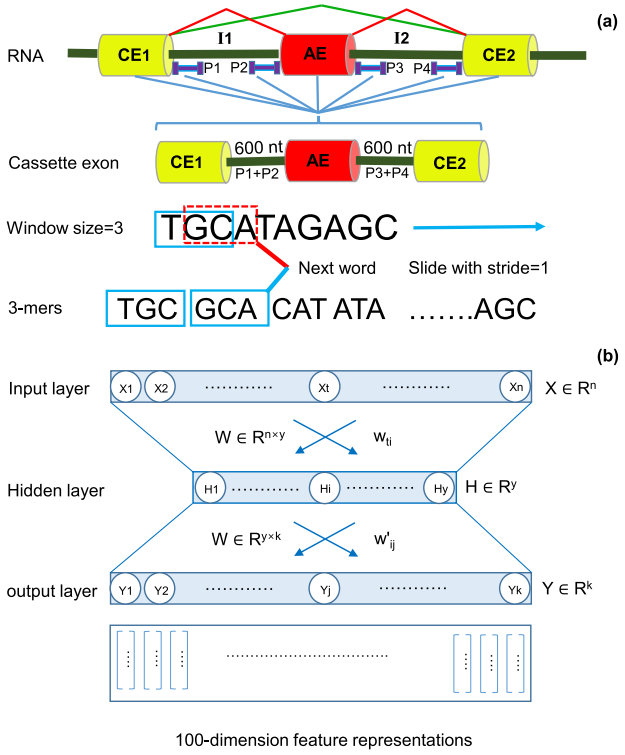

FIGURE 1. The procedure of data processing for distributed feature representations learning. (a) cassette exons formed from the alternative exon AE flanked by two constitutive exons CE1 and CE2 every two exons separated by introns, I1 between (CE1, AE) and I2 (AE, CE2). (b) the shallow neural network used to learn the feature representations.

B. DISTRIBUTED REPRESENTATION

In this work, we target the PSI prediction as a classification task where the output can be one of the three classes low, medium, or high. As well as a regression task where the PSI value is directly predicted. Mainly, the proposed approach comprises of two stages: feature learning and PSI prediction.

1) DISTRIBUTED FEATURE REPRESENTATIONS

In machine learning applications for genomic data analyses, it is common to extract a set of features for a range of tasks such as classification, regression, and clustering. Due to the complex and noisy nature of the raw genomic dataset. In this work, we decided to apply a feature representation learning technique [17]. This technique permits the generation of a more optimal set of features. Meanwhile, this step is also crucial to reduce the noise, decrease the computational complexity of the problem, and improve the performance of the computational models.

Word embedding or vector representation of words, its a crucial part of natural language processing (NLP) applications. Conceptually, it is a mathematical embedding from 1-dimension per word to continuous N-dimensional Vectors of real numbers. The growing interest in NLP over the past years because of its useful applications, such as speech recognition and translation devices, has led to a significant evolution in word embedding techniques.

Word2vec proposed by Mikolov *et al.* [18] is a neural network-based model that returns a continuously distributed representation for each word within a sentence based on the linguistic interaction between its words. Furthermore, in response to the need for similar representation for variable length texts (e.g., sentences, paragraphs, and documents) Le and Mikolov [19] proposed doc2vec. It is an extension of word2vec which represents each document as a fixed dimension vector regardless of its length.

It has been revealed that the genetic code can be considered as a language that passes the information within the cell and between cells [20]–[22]. This language is characterized by biological sequences such as DNA and RNA. Besides, applying NLP techniques to biological problems has been proved to be successful [21], [23]. Thus, we adopted the tow NLP frameworks word2vec and doc2vec to find interpretable representations for each cassette exons.

Unlike natural language text data, genome sequence data is a continuous chain of nucleotides based on four nucleobases Adenine (A), Cytosine (C), Guanine (G), Thymine (T). Also, it may contain a random nucleotide symbolized by "N" that could be interpreted as any of the four nucleotides. The first step in corpus construction, in this case, is to split the continuous biological sequence into a set of nucleotides of length k (k-mers) to break its continuity and form the words. The concept of words in NLP becomes k-mers. By referring to the previous works in which the effect of k length has been evaluated [24] and the concepts of overlapping and non-overlapping k-mers [24], [25] k was set to $k = 3$ with overlapping. We remove the regions with random nucleotides "N" which allows retention of the four nucleotides A, C, G, and T only. Now, each cassette exon contains a chain of continues nucleotides will be transformed to a sentence of overlapping words each of length 3 as illustrated in the following example. Subsequently, it will be processed using word2vec or doc2vec to generate the feature vectors.

For instance, a biological sequence ATGAACTG will result in the following sequence of words ATG TGA GAA AAC ACT CTG. The corpus consists of four different nucleotides forming words of length $k = 3$ and thus the vocabulary can contain $4^3 = 64$ maximum unique words. We have applied this pre-processing to construct a text corpus from the mouse genome, as described in the following section. This corpus is then used to train the word2vec and doc2vec.

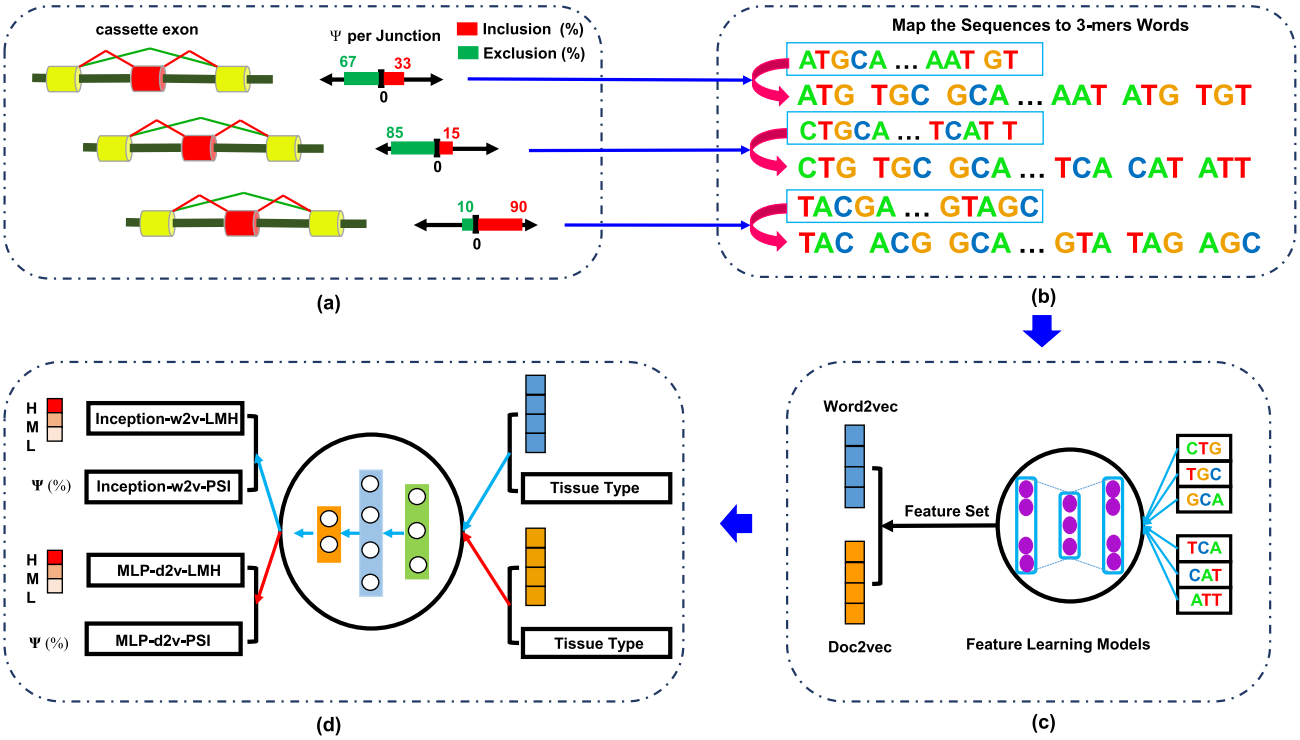

FIGURE 2. Schematic illustration of the proposed approach. (a) Quantification of RNA-Seq data and generate the PSI value (red) using MAJIQ. (b) Turn the continuous sequences to 3-mers forming the words. (c) feature representation from sequences using word2vec and doc2vec. (d) Deep learning based models for PSI prediction as classification and regression.

2) TRAIN word2vec AND doc2vec MODELS

Corpus preparation is a common procedure for both word2vec and doc2vec models training. We have generated the corpus by processing the whole mouse genome assembly (Release M16 version GRCm38.p5). It is obtainable from the GENCODE website:http://www.gencodegenes.org. Firstly, we have divided the genome assembly into 21 chromosomes (chr1, . . . , chr19, X, Y). Each chromosome was partitioned into sequences of 10,000 nt to form the sentences. Then we further broke down each sentence into overlapping 3-mers to form the words.

Word2vec model can be built based on one of the two available training methods: Continuous bag-of-words (CBOW) or skip-gram. Skip-gram executes the current word $w(t)$ to predict the surrounding window of context words. On the contrary, CBOW predicts the current word $w(t)$ based on the window of the surrounding context words. Given a window of size equal to 5, the input for CBOW based model is defined as follows:

$$\sum_{k=-2,k\neq 0}^{2} w(t+k). \tag{1}$$

The two methods described above perform similarly, however, skip-gram does a better job for infrequent words [26]. In this experiment, we essentially deal with frequent words. Thus, we have chosen CBOW over skip-gram as a training method for word2vec.

Likewise, doc2vec is proposed under two architectures: the distributed bag of words DBOW and distributed memory DM [18]. DBOW works in the same way as skip-gram, except that the input word is replaced by the document/sentence ID. On the other hand, DM works in a similar way to CBOW. In addition to the context words in the input, DM adds the documents ID. Fig. 3 illustrates both the CBOW and skip-gram architecture for word2vec model, besides DBOW and DM architecture for doc2vec model. The python library genism [27] was used to train these models. The selected training parameters are summarized in Table 1.

C. DEEP LEARNING MODELS

The previously proposed models are based on a simple architecture of BNN or DNN [10]–[12], [28]. These models mostly consist of two hidden layers as adding more layers do not improve the accuracy. Due to the new method proposed in this work for feature representation, which is completely different from the feature extraction method used in the previous works, re-implementation of these models was excluded. Seeing that convolution neural network (CNN) based models show high efficiency in computing range of core NLP tasks [29], we have proposed models based on Inception CNN architecture for PSI prediction. The proposed models consist of convolution layers. Each layer applies a convolution operation to the layer input, before passing the result to the next layer. In lieu of stacking layers on top of

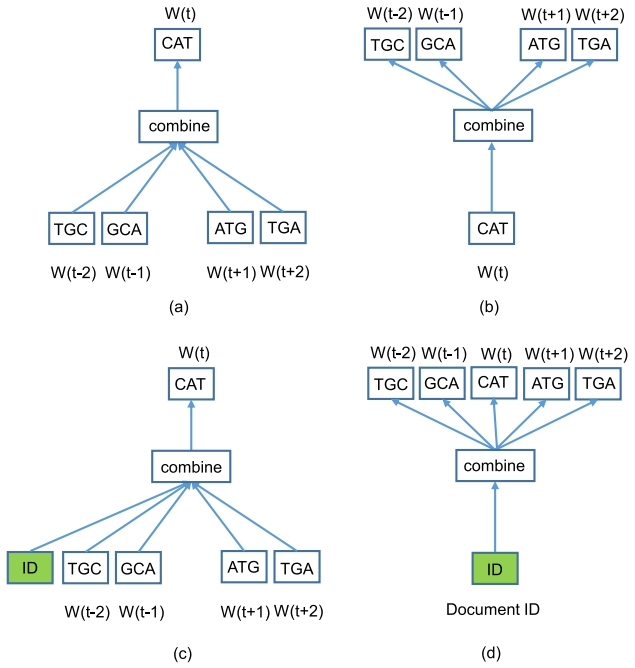

FIGURE 3. Different architectures for word2vec and doc2vec model. (a) CBOW for word2vec. (b) skip-gram for word2vec. (c) DM for doc2vec. (d) DBOW for doc2vec.

TABLE 1. Word2vec and doc2vec training parameters.

	Models	
Parameters	word2vec	doc2vec
Training Methode	CBOW	DM
Vector Size	100	100
Corpus	Mouse Genome	Mouse Genome
Context Words	3-mers	3-mers
Window Size	5	5
Minimum Count	5	3
Negative Sampling	5	5
Epochs	20	15

each other, we used the Inception architecture established in [30]. It allows the network to apply different convolutions with different kernel size in parallel, then concatenates the generated feature maps before going to the next block of layers. Also, inception architecture performs the dimensionality reduction. In fact, the additional 1×1 convolution used before the large convolutions (e.g., 3×3 and 5×5 convolutions are considered as large) performs the dimensionality reduction. The core inception block of the proposed models Inception-w2v-PSI and Inception-w2v-LMH is shown in Fig. 4.

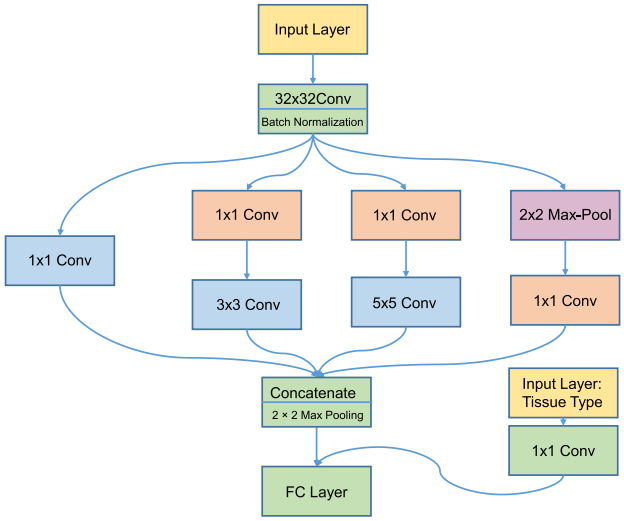

FIGURE 4. The core Inception model for both models Inception-w2v-PSI and Inception-w2v-LMH.

IMPLEMENTATION

The two frameworks word2vec and doc2vec, trained as described previously, were used to create the feature representation for each cassette exons in the dataset. Word2vec generate a 100-dimensional vector for every word in the sentence based on the context of the surrounding words. Whereas, doc2vec outputs one 100-dimensional vector for each sentence/document. These feature representations form the new input for different models instead of the cassette exons sequences. Additionally, the models take a second input concerning the tissue type. Since we process five tissues, a one-hot vector (1×5) is used to specify the tissue type for every input sample. For example, this input can be in this form [0 0 1 0 0] to point to the third tissue out of the five available tissues.

Inception-w2v-LMH and Inception-w2v-PSI are both based on one inception block and take as input the feature representations from the word2vec. Input with length less than the fixed input size is padded with vectors of zeros. The first layer applies 32×32 convolution. Followed by Batch normalization layer to normalize the features, by adjusting and scaling the activations. The output from this layer is passed to a single Inception module block, constituted of a variety of convolutions. We will be using 1×1, 3×3, 5×5 convolutions, along with a 2×2 max pooling and Rectified linear units (ReLU) as the activation function, as depicted in Fig. 4. Inception-w2v-PSI target function was set to directly predict the PSI value for each input cassette exons given its feature representation and tissue type. Whereas, Inception-w2v-LMH model target function was designed to map each input to one of the predefined class labels Low, Medium, or High.

Doc2vec generates a low dimensional vector of fixed size 100 for each cassette exons. Thus, we proposed an

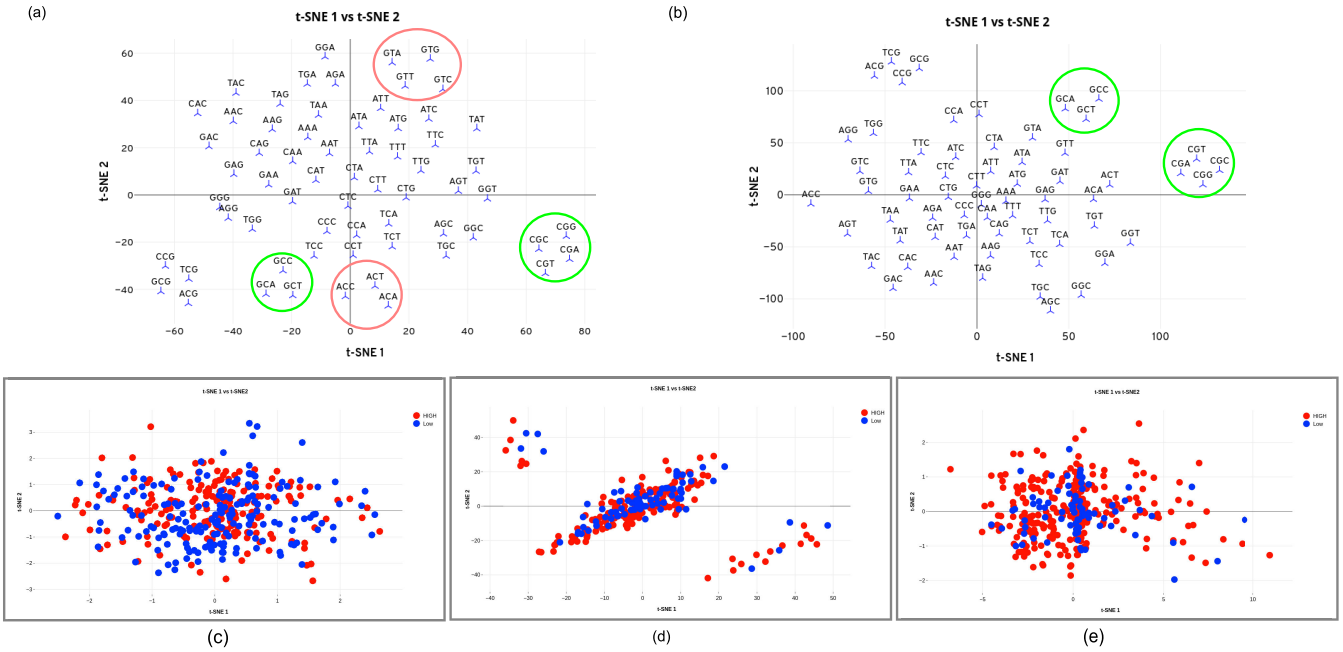

FIGURE 5. **Visualization of different plots using T-SNE: (a) word2vec vocabulary (b) doc2vec vocabulary (c) random embedding (d) embedding generated by word2vec (e) embedding generated by doc2vec. Each point represent a cassette exons from 2 classes, Low in blue and High in red.**

TABLE 2. **MLP architecture.**

Hyperparameter	Variants
Number of Hidden Layers	2
Number of Hidden Nodes	1024
Hidden Layers Activation function	ReLU
Dropout Keep Probability	0.8

architecture based on Multilayer Perceptron (MLP) neural network for both models that use the doc2vec feature vector. MLP is a class of feedforward artificial neural network. It consists of layers composed of nodes/neurons. In the proposed network structure neurons were assembled in two hidden layers, and only forward connection exists. Dropout regularization was applied to prevent the network from overfitting, and ReLU as an activation function. The output layer was customized to fit the two different tasks as follow: MLP-w2v-PSI model does the regression task. Consequently, in the output layer, the summation of outputs from hidden layers is weighted. Thence, passed through sigmoid activation function that gives the predicted value of PSI. MLP-w2v-LMH model performs the classification task. Therefore, a softmax activation function was used in the output layer to estimate the classes probabilities. MLP architecture implemented as shown in Table 2.

In the training process of each model, we followed the procedure proposed in [11] which is based on 5-fold cross-validation. Consequently, to train and test the four models the dataset was partitioned into five equal folds at random. Among these folds three used for training, one for validation, and one for testing. Training phase was monitored by early stopping. We have selected the hyper-parameters that give the optimal result on validation data then the final model was retrained in the training data and validation data together using the selected parameters. The evaluation is performed on the testing dataset. We perform three random permutations of the dataset to compute the standard deviation. The Keras Python library was used to build the models. We used a system of 6 units GPU NVIDIA GTX Titan X with total memory of 256 GB to accelerate the training process, especially for the word2vec and doc2vec training.

III. RESULTS

A. QUALITATIVE ANALYSIS OF word2vec AND doc2vec

In order to analyze and visualize the quality of the learned embeddings, we plotted the vectors assigned to all the 64 words in the vocabulary for both word2vec and doc2ved. As shown in Fig. 5 (first row). Also, the embeddings of a subset of the dataset from 2 classes low and high are displayed in Fig. 5 (second row). All the embedding were projected from 100-dimensional feature space into 2D space using t-distributed Stochastic Neighbor Embedding (t-SNE) [31]. In 2D space, it is notable that both word2vec and doc2vec were able to cluster words with similar properties. The most obvious examples are the clusters circled in

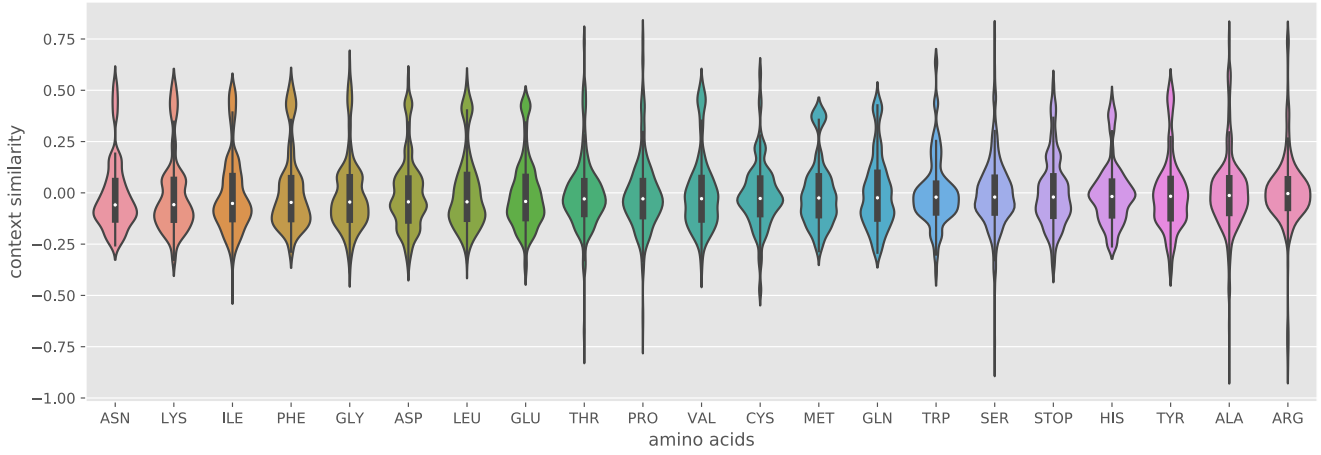

FIGURE 6. Violin plot represents the produced context similarity vectors for a sequence using word2vec. The x-axis represents the synonym codons classified based on the amino acid that they encode.

Fig. 5 (first row), and the commune property is the amino acid encoded: for word2vec the words cluster ACC, ACA and ACT encode Threonine. While, GTA, GTC, GTG, and GTT encodes Valine. For doc2vec the codons group GCA, GCT and GCC all encode the same amino acid Alanine. While CGT, CGC, CGG, and CGA encodes Arginine. We also notice that the word2vec and doc2vec have learned to gather the same clusters. As an example, the codons circled in green in Fig. 5 (first row).

Concerning evaluation on a subset of the dataset, we have used cassette exons from two classes low and high. Next, we generate the embeddings for each cassette exons using word2vec and doc2vec. For comparison reasons, we also generate an equal number of random 100-dimensional vectors labeled as low or high. Despite the overlapping between the points in the plots, due to the dimensionality reduction. The plots of word2vec and doc2vec show a better distribution allowing more contrast between the two classes compared to the random distribution. Moreover, Word2vec embeddings plot shows better partitioning of the two classes. Because word2vec model assigns a 100-dimensional vector for each word in the sentence, which allows capturing of more significant features.

Towards more illustration, we extend the analysis of similarity using the similarity measure function between two words available in genism library. A sequence from the data was used to compute the similarity of its codons to all synonym codons (i.e., encode for the same amino acid) using word2vec model. The distribution of context similarity for the different codons, which are classified based on the amino acids that they encode, is presented as a violin plot in Fig. 6. The median and interquartile range (IQR) are not very different among the different amino acids distributions. However, the shapes of these distributions are widely distinct. Concerning Threonine (THR), Proline (PRO), Serine (SER) and Alanine (ALA), the similarity distributions are of long similarity range compare to the rest of the amino acids. This because the number of synonym codons encodes these amino acids is more than four codons. The smallest distribution and similarity range belong to the Methionine (MET) meaning it has a few similar codons, as it encoded by a unique codon ATG.

We conclude from these analyses and visualizations that both word2vec and doc2vec captures many useful features. Moreover, the fundamental relations between codons/words maintained in a way that words with similar biological properties are clustered together. In 2D space, the generated representations contain important features. However, that will be more evident in the real 100-dimension space.

B. MODELS EVALUATION

In this work, we evaluate the models using two types of accuracy measure methods. In the case of the regression models, the predicted PSI value is compared to the estimated one from the RNA-seq quantification to compute the partition of variance explained (R^2). The area under the curve (AUC) is used to evaluate the performance of the classification models. Then we perform a comparison to the result reported in some previous works that have used the original dataset prepared by Brawand *et al.* [14].

In order to measure the effect of the new feature representation learning methods on the prediction accuracy, we assess MLP-d2v-LMH and Inception-w2v-LMH, both models perform the classification task. But MLP-d2v-LMH take as input the feature vectors generated using doc2vec, whereas Inception-w2v-LMH utilize those from word2vec. The target function used in the classification case has reduced the problem complexity, which has resulted in an overall similar performance of the two models across all the tissues. However, MLP-d2v-LMH model tend to achieve better performance than Inception-w2v-LMH model due to the different nature of the distributed representation given to

TABLE 3. Comparison of the classification task AUC performance on different methods.

Tissue	Method	Classes Low	Medium	High
Brain	MLR [11]	81.3 ±0.1	72.4 ±0.3	81.5 ±0.1
	BNN [11]	89.2 ±0.4	75.2±0.3	88.0 ±0.4
	DNN [11]	89.3 ±0.5	**79.4 ±0.9**	88.3 ±0.6
	Inception-w2v-LMH [ours]	92.2 ±0.1	72.8 ±0.6	**92.0 ±0.2**
	MLP-d2v-LMH [ours]	**93.0 ±0.4**	73.9 ±1.5	**92.8 ±0.3**
Heart	MLR [11]	84.6 ±0.1	73.1±0.3	83.6 ±0.1
	BNN [11]	91.1 ±0.3	74.7±0.3	89.5 ±0.2
	DNN [11]	90.7 ±0.6	**79.7 ±1.2**	89.4 ±1.1
	Inception-w2v-LMH [ours]	**96.5 ±0.2**	77.6 ±0.5	**96.2 ±0.4**
	MLP-d2v-LMH [ours]	**96.1 ±0.2**	77.3 ±1.0	95.8 ±0.1
Kidney	MLR [11]	86.7 ±0.1	75.6±0.2	86.3 ±0.1
	BNN [11]	92.5±0.4	78.3±0.4	91.6 ±0.4
	DNN [11]	91.9 ±0.6	**82.6 ±1.1**	91.2 ±0.9
	Inception-w2v-LMH [ours]	94.9 ±0.3	76.1 ±0.9	94.8 ±0.2
	MLP-d2v-LMH [ours]	**96.0 ±0.9**	80.1 ±1.3	**95.8 ±0.3**
Liver	MLR [11]	86.5 ±0.2	75.6 ±0.2	86.5 ±0.1
	BNN [11]	92.7 ±0.3	77.9±0.6	92.3 ±0.5
	DNN [11]	92.2 ±0.5	80.5 ±1.0	91.1±0.8
	Inception-w2v-LMH [ours]	96.8 ±0.7	78.7 ±1.3	96.6 ±0.8
	MLP-d2v-LMH [ours]	**97.1 ±0.5**	**90.9 ±0.8**	**97.0 ±0.6**
Testis	MLR [11]	85.6 ±0.1	72.3±0.4	85.2 ±0.1
	BNN [11]	**91.1 ±0.3**	75.5±0.6	**90.4 ±0.3**
	DNN [11]	90.7 ±0.6	**76.6 ±0.7**	89.7 ±0.7
	Inception-w2v-LMH [ours]	89.7 ±0.2	72.8 ±0.3	89.0 ±0.5
	MLP-d2v-LMH [ours]	89.2 ±0.3	73.5 ±0.9	89.3 ±0.5

± indicates standard deviation.

each model. Results are shown in Table 3. Further, We wished to compare the result of this task to the ones presented in [11], where three models based on different architectures MLR, BNN, and DNN were evaluated. As both the data and the code were not available to reproduce the result in [11] using the new RNA-quantification methods, we note here that the RNA-quantification methods and the selection criteria are different from those used in this work. Regardless of that, the proposed method achieves a better result for the four tissues (brain, kidney, heart, liver) and comparable result for the testis tissue, results are shown in Table 3. However the medium class ($0.33 \leq \Psi < 0.66$) is the most difficult class to predict, as only a small partition of a larger ratio of differential splicing event is grouped in this PSI range.

MLP-d2v-PSI and Inception-w2v-PSI models predict the inclusion level of the alternative exon PSI for each input cassette exons. MLP-d2v-PSI model is trained and tested on feature vectors from the doc2vec framework. Whereas, Inception-w2v-PSI use feature vectors from word2vec. Inception-w2v-PSI performs better than MLP-d2v-PSI in term of variance explained (R^2). As word2vec used in the first stage of MLP-d2v-PSI only computes a vector of size 100 for every document in the corpus. It does not generate a representative vector for each word in the document. In this case, some features are neglected and will not appears in the final distributed representations.

BNN-UDC and DNN-LMH were first introduced in [10] and later reconstructed by Jha *et al.* [12]. These models do not predict PSI directly. Thus, the authors compute the weighted average of the (low, medium, high) class prediction

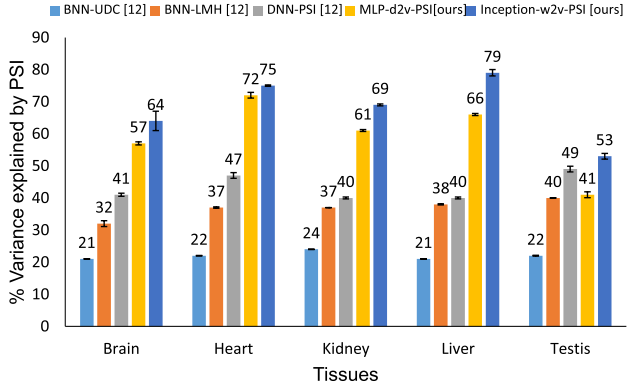

FIGURE 7. Improvement in explained variance achieved by the proposed models compares to the previous BNN and DNN models, the error bars are based on standard deviation.

probabilities. DNN-PSI model directly predicts PSI using the new target function introduced by [12]. These models use the same dataset in [14] and the same RNA-quantification method used in this work. The yellow and blue bars in Fig. 7 shows the significant improvement (12% – 39%) in the percent variance explained achieved by our models. The highest improvement 39% was achieved for the liver tissue.

IV. CONCLUSION

Building computer-based computational models (i.e., in silico biology) for complex biological phenomena, such as AS, helps to appreciate the hidden parameters that might not be in vitro accessible and reduce the cost and time of the experiments. In this work, we presented a novel method for learning distributed feature representation from the cassette exons, through the training of two NLP techniques word2vec and doc2vec. We provided experimental evidence showing that these techniques were able to learn valuable features from the biological sequences, also deciphering the different biological relations among the 3-mers/codons used as words. Next, these features used as input for different deep learning based models that accurately predict PSI. The proposed architectures outperformed the previous methods relies on features selection. Our ultimate goal is to discover the alternatively spliced genes linked to Alzheimer disease. As a future work, we will examine the application of these feature representation techniques in quantifying changes in splicing patterns across the tissues (delta PSI). We also aim to extend the dataset to include more tissue types.

REFERENCES

[1] D. L. Black, "Protein diversity from alternative splicing: A challenge for bioinformatics and post-genome biology," *Cell*, vol. 103, no. 3, pp. 367–370, 2000.
[2] Q. Pan, O. Shai, L. J. Lee, B. J. Frey, and B. J. Blencowe, "Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing," *Nature Genet.*, vol. 40, no. 12, pp. 1413–1415, 2008.
[3] Y. Wang *et al.*, "Mechanism of alternative splicing and its regulation," *Biomed. Rep.*, vol. 3, no. 2, pp. 152–158, 2015.
[4] E. T. Wang *et al.*, "Alternative isoform regulation in human tissue transcriptomes," *Nature*, vol. 456, no. 7221, pp. 470–476, 2008.

[5] M. R. Gazzara, J. Vaquero-Garcia, K. W. Lynch, and Y. Barash, "In silico to *in vivo* splicing analysis using splicing code models," *Methods*, vol. 67, no. 1, pp. 3–12, 2014.
[6] H. Y. Xiong *et al.*, "The human splicing code reveals new insights into the genetic determinants of disease," *Science*, vol. 347, no. 6218, p. 1254806, 2015.
[7] G. W. Yeo, E. L. Van Nostrand, and T. Y. Liang, "Discovery and analysis of evolutionarily conserved intronic splicing regulatory elements," *PLoS Genet.*, vol. 3, no. 5, p. e85, 2007.
[8] Q. Xu, B. Modrek, and C. Lee, "Genome-wide detection of tissue-specific alternative splicing in the human transcriptome," *Nucleic Acids Res.*, vol. 30, no. 17, pp. 3754–3766, 2002.
[9] Y. Barash *et al.*, "Deciphering the splicing code," *Nature*, vol. 465, no. 7294, pp. 53–59, 2010.
[10] H. Y. Xiong, Y. Barash, and B. J. Frey, "Bayesian prediction of tissue-regulated splicing using RNA sequence and cellular context," *Bioinformatics*, vol. 27, no. 18, pp. 2554–2562, 2011.
[11] M. K. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, 2014.
[12] A. Jha, M. R. Gazzara, and Y. Barash, "Integrative deep models for alternative splicing," *Bioinformatics*, vol. 33, no. 14, pp. i274–i282, 2017.
[13] Y. Barash *et al.*, "AVISPA: A Web tool for the prediction and analysis of alternative splicing," *Genome Biol.*, vol. 14, no. 10, p. R114, 2013.
[14] D. Brawand *et al.*, "The evolution of gene expression levels in mammalian organs," *Nature*, vol. 478, no. 7369, pp. 343–348, 2011.
[15] A. Dobin *et al.*, "STAR: Ultrafast universal RNA-seq aligner," *Bioinformatics*, vol. 29, no. 1, pp. 15–21, 2013, doi: 10.1093/bioinformatics/bts635.
[16] J. Vaquero-Garcia *et al.*, "A new view of transcriptome complexity and regulation through the lens of local splicing variations," *Elife*, vol. 5, p. e11752, Feb. 2016.
[17] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (Jan. 2013). "Efficient estimation of word representations in vector space." [Online]. Available: http://arxiv.org/abs/1301.3781
[19] Q. V. Le and T. Mikolov. (May 2014). "Distributed representations of sentences and documents." [Online]. Available: http://arxiv.org/abs/1405.4053
[20] D. B. Searls, "String variable grammar: A logic grammar formalism for the biological language of DNA," *J. Log. Program.*, vol. 24, nos. 1–2, pp. 73–102, 1995.
[21] M. D. Yandell and W. H. Majoros, "Genomics and natural language processing," *Nature Rev. Genet.*, vol. 3, pp. 601–610, Aug. 2002, doi: 10.1038/nrg861.
[22] C. Emmeche and J. Hoffmeyer, "From language to nature: The semiotic metaphor in biology," *Semiotica*, vol. 84, nos. 1–2, pp. 1–42, 1991.
[23] K. B. Cohen and L. Hunter, "Natural language processing and systems biology," in *Artificial Intelligence Methods And Tools For Systems Biology*. Dordrecht, The Netherlands: Springer, 2004, pp. 147–173.
[24] P. Ng. (2017). "dna2vec: Consistent vector representations of variable-length k-mers." [Online]. Available: https://arxiv.org/abs/1701.06279
[25] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS One*, vol. 10, no. 11, pp. 1–15, 2015, doi: 10.1371/journal.pone.0141287.
[26] L. Recalde, J. Mendieta, L. Boratto, L. Teran, C. Vaca, and G. Baquerizo, "Who you should not follow: Extracting word embeddings from tweets to identify groups of interest and hijackers in demonstrations," *IEEE Trans. Emerg. Topics Comput.*, to be published.
[27] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proc. LREC Workshop New Challenges NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50. [Online]. Available: http://is.muni.cz/publication/884893/en
[28] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[29] X. Ouyang, P. Zhou, C. H. Li, and L. Liu, "Sentiment analysis using convolutional neural network," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. Ubiquitous Comput. Commun. Dependable, Autonomic Secure Comput. Pervasive Intell. Comput.*, Oct. 2015, pp. 2359–2364.
[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.
[31] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

MHANED OUBOUNYT received the B.Sc. degree in informatics and electronics engineering from the Cadi Ayyad University of Sciences and Technology, Marrakech, Morocco, in 2016. He is currently pursuing the M.Sc. degree with the Department of Electronics and Information Engineering, Chonbuk National University, Jeonju, South Korea, with a focus on applying machine learning methods in bioinformatics. His research interests are focused on artificial intelligence, deep learning, machine learning, image processing, and medical engineering.

ZAKARIA LOUADI received the B.Sc. degree in informatics and electronics engineering from Cadi Ayyad University, Marrakesh, Morocco, in 2016. He is currently pursuing the master's degree in electronics and information Chonbuk National University, Jeonju, South Korea. His research interests include machine learning, deep learning, and computer vision and currently focused on using computational models to identify novel regulatory mechanisms of alternative splicing.

HILAL TAYARA received the B.Sc. degree in computer engineering from Aleppo University, Aleppo, Syria, in 2008, and the M.S. degree in electronics and information engineering from Chonbuk National University, Jeonju, South Korea, in 2015. He is currently a Researcher with Chonbuk National University. His research interests include machine learning and image processing.

KIL TO CHONG received the Ph.D. degree in mechanical engineering from Texas A&M University in 1995. He is currently a Professor with the School of Electronics and Information Engineering, Chonbuk National University, Jeonju, South Korea, and the Head of the Advanced Research Center of Electronics. His research interests are in the areas of machine learning, signal processing, motor fault detection, network system control, and time-delay systems.