Title

DeepPASTA: deep neural network based polyadenylation site analysis

Journal

Bioinformatics, 35(22)

ISSN

1367-4803

Authors

Arefeen, Ashraful

Xiao, Xinshu

Jiang, Tao

Publication Date

2019-11-01

DOI

10.1093/bioinformatics/btz283

Peer reviewed

Bioinformatics, 35(22), 2019, 4577–4585
doi: 10.1093/bioinformatics/btz283
Advance Access Publication Date: 25 April 2019
Original Paper

OXFORD

Genome analysis

DeepPASTA: deep neural network based polyadenylation site analysis

Ashraful Arefeen [1], Xinshu Xiao[2,*] and Tao Jiang[1,3,4,*]

[1]Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA, [2]Department of Integrative Biology and Physiology, University of California, Los Angeles, CA 90095, USA, [3]Institute of Integrative Genome Biology, University of California, Riverside, CA 92521, USA and [4]Bioinformatics Division, BNRIST, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on November 21, 2018; revised on March 22, 2019; editorial decision on April 11, 2019; accepted on April 16, 2019

Abstract

Motivation: Alternative polyadenylation (polyA) sites near the 3′ end of a pre-mRNA create multiple mRNA transcripts with different 3′ untranslated regions (3′ UTRs). The sequence elements of a 3′ UTR are essential for many biological activities such as mRNA stability, sub-cellular localization, protein translation, protein binding and translation efficiency. Moreover, numerous studies in the literature have reported the correlation between diseases and the shortening (or lengthening) of 3′ UTRs. As alternative polyA sites are common in mammalian genes, several machine learning tools have been published for predicting polyA sites from sequence data. These tools either consider limited sequence features or use relatively old algorithms for polyA site prediction. Moreover, none of the previous tools consider RNA secondary structures as a feature to predict polyA sites.

Results: In this paper, we propose a new deep learning model, called DeepPASTA, for predicting polyA sites from both sequence and RNA secondary structure data. The model is then extended to predict tissue-specific polyA sites. Moreover, the tool can predict the most dominant (i.e. frequently used) polyA site of a gene in a specific tissue and relative dominance when two polyA sites of the same gene are given. Our extensive experiments demonstrate that DeepPASTA signisficantly outperforms the existing tools for polyA site prediction and tissue-specific relative and absolute dominant polyA site prediction.

Availability and implementation: https://github.com/arefeen/DeepPASTA

Contact: gxxiao@ucla.edu or jiang@cs.ucr.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

According to the central dogma of molecular biology, the genomic sequence of an eukaryotic gene is transformed into the corresponding protein by the transcription, post-transcriptional and translation processes. Initially, the transcription process converts a gene into a pre-mRNA, then this pre-mRNA is transformed into a mature mRNA by the post-transcriptional process and finally the mRNA is translated into the corresponding protein by the translation process. One of the important steps of the post-transcriptional process is the addition of a polyadenylation (polyA) tail at the 3′ end of a pre-mRNA. More specifically, the polyadenylation process consists of two steps (Wahle and Kühn, 1997): cleavage near the 3′ end of a pre-mRNA and addition of a polyA tail at the cleavage site or polyA site.

Alternative cleavage sites near the 3′ end of a pre-mRNA create more than one mRNA transcript containing 3′ untranslated regions (3′ UTRs) of different lengths. A 3′ UTR is a suffix of an mRNA sandwiched between the stop codon and polyA site of the mRNA.

The length of a 3′ UTR as well as some sequence elements (such as those AU and GU rich elements) may have impact on mRNA stability, mRNA localization, protein translation, protein binding and translation efficiency (Barrett *et al.*, 2012). For example, longer 3′ UTRs may have additional destabilization elements that alter the respective transcript's stability (Shaw and Kamen, 1986) and in cancers, transcripts with shorter 3′ UTR can escape regulation from microRNAs (Di Giammartino *et al.*, 2011; Lin *et al.*, 2012). Moreover, the secondary structure of a 3′ UTR is also important for translation efficiency and disruption of expression (Barrett *et al.*, 2012). Alternative polyadenylation is very common in mammalian genes (Tian *et al.*, 2005) and more than a half of human genes have alternative polyadenylation in their post-transcription process (Mayr, 2016). Moreover, errors in 3′-end processing may cause several inherited diseases (Danckwardt *et al.*, 2008). Therefore, the prediction and analysis of polyA sites would be of great importance in the study of mammalian genes.

Several cis-elements and trans-factors influence the choice of a polyA or cleavage site (Barrett *et al.*, 2012; Pichon *et al.*, 2012). The most important cis-element for a polyA site is the hexamer or polyadenylation signal (PAS), which usually occurs 10–30 nt upstream of the cleavage site (Akhtar *et al.*, 2010; Bajic *et al.*, 2012; Cheng *et al.*, 2006; Derti *et al.*, 2012; Liu *et al.*, 2005; Salamov *et al.*, 1997; Tabaska *et al.*, 1999; Tian *et al.*, 2005; Yada *et al.*, 1994). The PAS serves as a binding site for the cleavage and polyadenylation specificity factor (CPSF) (Colgan *et al.*, 1997). A polyA site also depends on the U or U/G-rich elements and these elements occur 20–40 nt downstream of that polyA site (Akhtar *et al.*, 2010; Cheng *et al.*, 2006; Derti *et al.*, 2012; Tian *et al.*, 2005). These U or U/G-rich elements serve as the binding sites for the cleavage stimulation factor (CstF) (Colgan *et al.*, 1997). In addition, some auxiliary elements upstream of the PAS and downstream of the cleavage site may enhance the polyadenylation process (Akhtar *et al.*, 2010; Hu, 2005; Tian *et al.*, 2005). Therefore, a polyA site typically depends on four different cis-elements: auxiliary upstream elements (AUEs), the upstream hexamer signal (i.e. PAS), downstream U/GU rich elements and auxiliary downstream elements (ADEs). Moreover, the RNA secondary structures near the downstream region of mammalian PASs impact the choice of polyA sites (Brown *et al.*, 1991; Wu *et al.*, 2004).

Several tools have been introduced in the literature to predict polyA sites or PASs from human genomic sequences. DNAFSMiner (Liu *et al.*, 2003; Liu *et al.*, 2005) predicts PASs from sequences using k-mer features in a support vector machine (SVM) model. Dragon PolyA Spotter (Bajic *et al.*, 2012) also predicts PASs from sequences using both an artificial neural network and a random forest. POLYAH (Salamov *et al.*, 1997) discriminates real PASs from other hexamer signals using a linear discriminant function. It focuses on only one PAS (AATAAA) in its analysis, although other PASs (variants of AATAAA) may influence polyA site selection. Polyadq (Tabaska *et al.*, 1999) uses a quadratic linear discriminant function to predict real PAS regions. This tool considers only two signals (A(A/T)TAAA) in its analysis. However, a polyA site not only depends on the upstream PAS but also downstream U/GU rich elements, AUEs and DAEs. Polya_svm (Cheng *et al.*, 2006) predicts polyA sites from sequences using a SVM model. PolyAR (Akhtar *et al.*, 2010) also predicts polyA sites from sequences using a linear discriminant function. However, these tools use hand-picked sequence features. In order to overcome the limitation of hand-picked sequence features, deep learning models such as DeepPolyA (Gao *et al.*, 2018), DeeReCT-PolyA (Xia *et al.*, 2018) and Conv-Net (Leung *et al.*, 2018) have been recently introduced to predict polyA sites, PASs and relatively dominant polyA sites (i.e. more frequently used polyA sites in a given gene). These models use all convolution neural networks (CNNs) to extract features from the input genomic sequence. Although the secondary structure near a polyA site is essential for the polyA site to be selected for the polyadenylation process (Bar-Shira *et al.*, 1991; Brown *et al.*, 1991; Wu *et al.*, 2004), none of these tools consider RNA secondary structures in their prediction procedures.

Polyadenylation occurs in a tissue specific manner (Hafez *et al.*, 2013; Tian and Manley, 2017; Weng *et al.*, 2016; Zhang *et al.*, 2005). Different tissues show bias in selecting the locations of polyA sites within a gene, such as sites located in introns, internal exons and the last exon (Zhang *et al.*, 2005). Different polyA sites in the last exon may result in mRNAs with different 3′ UTRs. On the other hand, the usage of polyA sites located in introns or internal exons may lead to the creation of premature stop codons or truncated proteins. Therefore, predicting the locations of tissue-specific polyA sites is important for understanding tissue-specific behaviors, variable 3′ UTRs and protein products (Zhang *et al.*, 2005).

One way to study tissue-specific choices of polyA sites is to consider the usages of different polyA sites. In a last exon, polyA sites closest to the 5′ and 3′ ends are called the proximal and distal polyA sites, respectively (Zhang *et al.*, 2005). There are other polyA sites in between proximal and distal polyA sites, and these sites are called middle polyA sites (Zhang *et al.*, 2005). Placenta, retina, blood, testis and ovary tissues show preference for proximal polyA sites, i.e. high usage of proximal and low usage of distal polyA sites. On the other hand, bone marrow, uterus, ear, brain, the nervous system and pancreatic islet show high usage of distal polyA sites (Zhang *et al.*, 2005). Therefore, it would be interesting to predict relatively dominant polyA sites for a given gene to understand tissue specific behaviors. Conv-Net (Leung *et al.*, 2018) is the first published tool to analyze relative dominance of polyA sites in human tissues. More specifically, the tool takes a couple of polyA sites within a 3′ UTR and predicts the dominant polyA site using a deep learning algorithm.

In this paper, we introduce a new tool, called DeepPASTA (i.e. Deep neural network-based PolyA SiTe Analysis), to predict polyA sites from sequences and RNA secondary structures. As secondary structure near a polyA site is important for the polyA site selection (Brown *et al.*, 1991; Wu *et al.*, 2004), DeepPASTA is the first tool to consider both sequence and RNA secondary structure in polyA site prediction. It employs both a CNN and a recurrent neural network (RNN). The CNN extracts features from sequences (Alipanahi *et al.*, 2015; Angermueller *et al.*, 2017; Kelley *et al.*, 2016; Zhou *et al.*, 2015) and secondary structures. On the other hand, the RNN is used to combine the effects of upstream and downstream signals (Colgan *et al.*, 1997; Wahle *et al.*, 1995) in polyA site prediction. As the polyadenylation process is tissue-specific, we also formulate tissue-specific polyA site prediction as a multi-label classification problem where the usage of a polyA site is simultaneously analyzed for multiple tissues, and extend DeepPASTA to solve this problem. Similar to Conv-Net (Leung *et al.*, 2018), DeepPASTA can also predict relatively dominant polyA sites of a gene in a specific tissue. We further generalize the relative dominance problem so DeepPASTA can also predict the most dominant polyA sites for each gene (i.e. the absolute dominance problem).

To assess the performance of DeepPASTA, we have conducted extensive experiments on human genomic sequence data and compared DeepPASTA with the above mentioned tools including PolyAR, Dragon PolyA Spotter, DeepPolyA, DeeReCT-PolyA and Conv-Net for polyA site prediction or relative dominance. As none

of the existing tools are able to perform all four types of polyA site analysis that DeepPASTA can do, we organize the comparisons as four groups: (i) prediction of human polyA sites (between DeepPASTA, PolyAR, Dragon PolyA Spotter, DeepPolyA, DeeReCT-PolyA and Conv-Net), (ii) prediction of human tissue-specific polyA sites (between DeepPolyA, the tissue-specific and non-tissue-specific DeepPASTA models), (iii) prediction of relatively dominant polyA sites (between DeepPASTA and Conv-Net) and (iv) prediction of absolute dominant polyA sites (between DeepPASTA and Conv-Net). The tools are compared in term of area under the curve (AUC) and area under the precision recall curve (AUPRC). Based on these two performance measures, DeepPASTA outperforms the other tools significantly in polyA site prediction. For tissue-specific relatively dominant polyA site prediction, DeepPASTA achieves better AUCs and AUPRCs than Conv-Net most of the human tissues. In tissue-specific absolute dominant polyA site prediction, DeepPASTA again outperforms Conv-Net on all human tissues.

The rest of the paper is organized as follows. The four different models of DeepPASTA are discussed in Section 2. The experimental results and comparisons with the other tools are discussed in Section 3. Section 3 also explains the sequence and secondary structure data generation procedure. At the end, conclusion is drawn in Section 4.

2 Materials and methods

In this section, we describe the four models of DeepPASTA for predicting polyA sites, tissue-specific polyA sites, tissue-specific relative dominance between polyA sites, and tissue-specific absolutely dominant polyA sites from human genomic sequence and RNA secondary structure data. The four prediction problems are formally defined as follows. The first problem is a binary classification problem that takes a genomic sequence of 200 nt (Akhtar *et al.*, 2010; Leung *et al.*, 2018) and some probable secondary structures predicted by RNAshapes (Steffen *et al.*, 2006) as the input and expects a score as the output indicating the likelihood for the middle position of the input sequence to be a polyA site. Note that RNAshapes is used here because it is one of the most popular tools for RNA secondary structure prediction (Maticzka *et al.*, 2014; Zhang *et al.*, 2015). The second problem is a multi-label classification problem that takes a sequence and some corresponding RNA secondary structures as the input and asks which tissues may have polyA sites in the input sequence for a given set of tissues. The third problem in a multi-class classification problem that takes two polyA sites surrounding sequences (200 nt) as well as corresponding RNA secondary structures of a gene as the input and estimates the relatively dominant polyA site in a particular tissue. The final problem is a binary classification problem that takes a polyA site and its surrounding sequence (200 nt) as well as corresponding RNA secondary structure of a gene as the input and outputs a score indicating the likelihood for the input polyA site to be the absolutely dominant polyA site of the gene. The detailed input and output of the above four models are illustrated in Supplementary Figure S1.

Recently, deep learning has been applied in bioinformatics with superior performance over conventional learning methods on many prediction/classification problems, such as protein-nucleotide binding prediction (Alipanahi *et al.*, 2015; Pan and Shen, 2017; Zhang *et al.*, 2015), functional genomic data prediction (Eser and Churchman, 2016), translation initiation site (Zhang *et al.*, 2017a) and ribosome stalling prediction (Zhang *et al.*, 2017b). Following these state-of-the-art methods, DeepPASTA also uses deep learning algorithms in its prediction models. Each of the four models of DeepPASTA employs both a CNN for extracting features and an RNN for combining the effects of these features. The four models are explained in detail in Sections 2.1–2.4.

2.1 Predicting polyA sites

The first model of DeepPASTA is the polyA site prediction model. The model takes a genomic sequence of 200 nt and some corresponding RNA secondary structures as the input to predict whether that sequence has a polyA site in the middle or not. Following the literature (Zhang *et al.*, 2015), three energy efficient RNA secondary structures are generated by RNAshapes (Steffen *et al.*, 2006) from the sequence and given as a part of the input. Supplementary Figure S2 shows the overall architecture of the model. The model consists of four sub-models: a sequence sub-model and three identical secondary structure sub-models. The sequence sub-model starts with a convolution layer (LeCun *et al.*, 1998). Following the work (Alipanahi *et al.*, 2015; Angermueller *et al.*, 2017; Kelley *et al.*, 2016; Zhou *et al.*, 2015), the convolution layer of DeepPASTA is used to extract features from the input sequence based on a sliding window. It uses a rectified linear unit (ReLU) (Nair and Hinton, 2010) as the activation function to set negative values to zero. The next layer is a max pooling layer (Ciregan *et al.*, 2012) that picks the maximum feature value within a window. After the max pooling layer, a bidirectional LSTM (long short term memory) recurrent layer (Gers *et al.*, 2000; Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997) is used to consider both upstream and downstream signals for polyA site prediction. The last layer of the sequence sub-model is a fully connected layer. Each of the three input RNA secondary structures is fed to a secondary structure sub-model. Similar to the sequence sub-model, each secondary structure sub-model starts with a convolution layer to extract features from the input secondary structure. This convolution layer is followed by an average pooling layer. The average pooling layer calculates the average of all the feature values within a window. The next two layers of the sub-model are a bidirectional LSTM and a fully connected layer. The three secondary structure sub-models are combined using an addition layer and then concatenated with the sequence sub-model. The concatenation layer is followed by multiple fully connected layers. The fully connected layers of the polyA site prediction model use ReLUs as the activation function. The model ends with a single neuron output layer with a sigmoid activation function. In order to prevent data overfitting, dropouts (Srivastava *et al.*, 2014) are used in some of the layers.

Separate training and validation data are used to train the model, while some test data is used to evaluate the performance of the trained model. Ground truth values of the training and validation data are taken from the PolyA-Seq data in Derti *et al.* (2012). The PolyA-Seq data provide tissue specific polyA sites in human. For the basic (i.e. non-tissue specific) polyA site prediction problem, we take the union of all the tissues to construct the ground truth data. Supplementary Figure S3 shows the steps of the training phase. The model is trained using the Adam RMSprop with Nesterov momentum (Dozat, 2016) optimizer. It uses a minibatch size of 1000 to minimize the average multi-task binary cross entropy loss on the training data. At the end of each training epoch, the validation loss is evaluated to monitor convergence. In order to expedite the learning process, a graphic processing unit (GPU) is used.

2.2 Predicting tissue-specific polyA sites

If a sequence and its corresponding RNA secondary structures are given as the input, the tissue-specific polyA site prediction model

asks in which tissues the sequence contains a polyA site. The model is a multi-label classifier (Aly, 2005) that simultaneously considers nine different human tissues: brain, kidney, liver, maqc_brain1, maqc_brain2, maqc_UHR1, maqc_UHR2, muscle and testis (Derti *et al.*, 2012). Following the literature (Leung *et al.*, 2018), we consider the four samples maqc_brain1, maqc_brain2, maqc_UHR1 and maqc_UHR2 as four different tissues. The output from the multi-label classifier is a vector of nine values for the nine tissues. If a tissue has a polyA site in the input sequence then the tissue's corresponding value in the vector should be 1, otherwise 0. Note that the model also works for input sequence that does not contain any polyA site for the nine tissues. Here, our model considers nine tissues because the ground truth data, PolyA-Seq (Derti *et al.*, 2012), involve nine tissues, but it can be applied to any set of tissues that has ground truth data.

Supplementary Figure S4 shows the overall architecture of the model. Similar to the basic (i.e. non-tissue-specific) polyA site prediction model described in Section 2.1, this model has one sequence and three identical secondary structure sub-models. The sequence and secondary structure sub-models are similar to the sequence and secondary structure sub-models of the basic polyA site prediction model, but they use parametric ReLUs (PReLU) (He *et al.*, 2015) as the activation functions in its convolution and fully connected layers. The four sub-models are then combined using two concatenation layers. The latest concatenation layer is followed by multiple fully connected layers and these fully connected layers use ReLUs as the activation functions. The final layer of the model has nine output neurons for the nine tissues and it uses a sigmoid activation function. Dropouts are used in some of the layers to prevent overfitting.

The training process of this model is similar to the training process of the basic polyA site prediction model.

2.3 Predicting tissue-specific relatively dominant polyA sites

When a couple of sequences of 200 nt (and corresponding RNA secondary structures) containing polyA sites of some gene in a particular tissue are given as the input, this model predicts which polyA site is more dominant in the tissue [i.e. more frequently used; (Leung *et al.*, 2018)]. Note that we do not define this model as a multi-label classifier because the input sequence may not contain polyA sites in all tissues. Supplementary Figure S5 shows the overall architecture of the model. The (relative dominance) strength value of each input sequence (and RNA secondary structure) is calculated using a sub-unit of the model. This sub-unit consists of two sub-models: sequence and secondary structure sub-models. The sequence sub-model consists of one convolution layer with a PReLU activation function, one max pooling layer, one bidirectional LSTM recurrent layer and one fully connected layer with a ReLU activation function. The architecture of the secondary structure sub-model is similar to the sequence sub-model, but uses an average pooling layer in place of the max pooling layer. The sequence and secondary structure sub-models are combined using a concatenation layer. The concatenation layer is followed by multiple fully connected layers. The sub-unit ends with an output layer with a single neuron that provides the strength value of the input sequence. At the end, the relatively more dominant polyA site is determined by comparing the output strength values of the two input sequences from the two sub-units using softmax (Bishop, 2006).

In order to train the model, the read counts of the two input sequences from the PolyA-Seq data (Derti *et al.*, 2012) are used to construct the ground truth. More specifically, if the input sequences S1 and S2 have R_1 and R_2 read counts, then the ground truth strengths of these two sequences are $\frac{1+R_1}{(2+R_1+R_2)}$ and $\frac{1+R_2}{(2+R_1+R_2)}$, respectively. A similar procedure is also followed to calculate the ground truth in Conv-Net (Leung *et al.*, 2018). The training process of the relative dominant polyA site prediction model is similar to the training process of the polyA site prediction model.

2.4 Predicting tissue-specific absolutely dominant polyA sites

When a sequence of 200 nt (and corresponding RNA secondary structure) containing a polyA site of some gene in a particular tissue is given as the input, this model predicts whether the polyA site is a most dominant site (i.e. the most frequently used) of the gene or not in the involved tissue. Usually, the most dominant polyA sites of a gene are more likely selected for the polyadenylation process. Again, we do not define this model as a multi-label classifier because the input sequence may not contain polyA sites in all tissues. Supplementary Figure S6 shows the overall architecture of the model. The absolutely dominant prediction model has two sub-models: sequence and secondary structure sub-models. The sequence and secondary structure sub-models of this model are similar to the sequence and secondary structure sub-models of the relative dominance prediction model. These two sub-models are also combined using a concatenation layer. The concatenation layer is followed by multiple fully connected layers with ReLU activation functions. The final layer of the model has one output neuron and the activation function of this layer is sigmoid.

The read count values of polyA sites from the PolyA-Seq data are used to determine one or more absolutely dominant polyA sites of a gene. PolyA sites with the maximum read counts within a gene are considered as the (absolutely) dominant polyA sites and rest are considered as the non-dominant sites. The model is trained using similar steps as the other models described above.

3 Experimental results

In this section, we compare the performance of DeepPASTA with that of some state-of-the-art methods for predicting polyA sites, tissue-specific polyA sites as well as relatively/absolutely dominant polyA sites. We also compare the tissue-specific polyA site prediction model with the non-tissue-specific model.

In order to construct the sequence data for DeepPASTA's models, polyA sites are collected from the PolyA-Seq experiments in Derti *et al.* (2012). As AUEs, the PAS, U/GU rich elements and ADEs are typically within 100 nt upstream and downstream of a polyA site (Akhtar *et al.*, 2010; Hu, 2005), the genomic sequence of length 200 nt centered around a polyA site is taken from the human GRCh37 (hg19) reference genome (similar to Leung *et al.*, 2018). These sequences are considered as the positive examples for the deep learning models. The models also need negative examples for training and testing. Therefore, four different sets of negative examples are constructed: two sets obtained by shifting each positive example left and right by 50 nt (Leung *et al.*, 2018), random sequences containing upstream hexamer signals, and random sequences from coding and noncoding regions of genes (Akhtar *et al.*, 2010). The length of each of these negative examples is 200 nt. In the shifted negative examples, the polyA sites are not in the middle of the 200 nt sequences. In the negative examples with hexamer signals, the hexamer signals are in the upstream region of the sequences. Similar to the literature (Gao *et al.*, 2018; Leung *et al.*, 2018; Zhang *et al.*, 2017a), these sequence examples are then fed to DeepPASTA by using the

one-hot encoding representation. As there are four possible nucleotides (A, C, G and T) in DNA sequences, the dimensionality of a sequence example is 4×200.

Since the selection of polyA sites in polyadenylation process also depends on the RNA secondary structures near the polyA sites (Brown *et al.*, 1991; Wu *et al.*, 2004). DeepPASTA considers both sequences and their corresponding RNA secondary structures for its prediction tasks as mentioned above. RNAshapes (Steffen *et al.*, 2006) is used to predict the probable secondary structures of each sequence example (Maticzka *et al.*, 2014; Zhang *et al.*, 2015). More specifically, the sequence is scanned using a sliding window (of size 100 nt) and a step size (of 100 nt) to predict first level abstract (i.e. the most detailed) representation/secondary structures (Lange *et al.*, 2012). As done in Zhang *et al.* (2015), the three most energy efficient secondary structures of the sequence are recorded for future analyses. Each position of the secondary structure is represented by one of seven symbols, i.e. L, R, U, M, H, I and E, which stand for left hand base of a double strand, right hand base of a double strand, unpaired base, multiloop, hairpin loop, internal loop and external region, respectively. Similar to the sequence input, the secondary structure input uses a one-hot encoding representation. Therefore, the dimensionality of a secondary structure input is 7×200.

3.1 Performance on predicting polyA sites

In this experimental study, we compare DeepPASTA with five existing tools: PolyAR (Akhtar *et al.*, 2010), Dragon PolyA Spotter (Bajic *et al.*, 2012), DeeReCT-PolyA (Xia *et al.*, 2018), Conv-Net (Leung *et al.*, 2018) and DeepPolyA (Gao *et al.*, 2018) for predicting polyA sites on three datasets (to be introduced below). In order to train the model of DeepPASTA, we partition the human chromosomes into three groups: chromosomes 1 to 8 as group 1, chromosomes 9 to 14 as group 2 and chromosomes 15 to Y as group 3. Homologous genes from BioMart of Ensembl are considered to prevent potential data leak (i.e. training data containing information of test data). Using the genes from chromosomes 1 to 8, homologous genes are extracted from chromosomes 9 to Y and are added into group 1. Similarly, using the genes from chromosomes 9 to 14, homologous genes are extracted from chromosomes 15 to Y and are added into group 2. The polyA sites in groups 1, 2 and 3 are collected from the PolyA-Seq data. Before homologous genes are moved, group 1, 2 and 3 have 251 726, 125 665 and 144 208 polyA sites, respectively. After moving homologous genes, the number of polyA sites in group 1, 2 and 3 are 326 695, 99 498 and 95 406, respectively. We then construct training, validation and test data from groups 1, 2 and 3, respectively. As mentioned above, four different types of negative sequences and their corresponding RNA secondary structures are collected from group 1 and down-sampled to make the ratio of positive and negative example as 1 : 1 in the training data. The down-sampling helps make the prediction model more robust (Akhtar *et al.*, 2010; Bajic *et al.*, 2012; Liu *et al.*, 2003). A similar procedure is also followed to construct the validation data. The polyA site prediction model of DeepPASTA is then trained using the training and the hyperparameters of the model are tuned empirically using held-out validation data. The test data are constructed similarly, but we do not down-sample the negative examples (thus keeping the ratio of positive and negative examples as 1:4).

We use three test datasets to evaluate the performance of the tools. Datasets 1 and 2 are constructed from the test data and dataset 3 is taken from the literature (Leung *et al.*, 2018). Dataset 1 is constructed using the whole test data. Therefore, it contains 95 406 positive and 381 555 negative examples (ratio 1 : 4). Dataset 2 is a subset of dataset 1 and it consists of 95 406 (i.e. all) positive examples and 95 406 random sequences as the negative examples. We include both balanced data (dataset 2) and unbalanced data (dataset 1) in the performance evaluation because PolyAR and Dragon PolyA Spotter use balanced datasets in their performance evaluations (Akhtar *et al.*, 2010; Bajic *et al.*, 2012) but in reality, the number of polyA sites is very small compared to the whole human genome. Note that datasets 1 and 2 do not contain any information about the training and validation data.

In order to compare with the most recent method Conv-Net (Leung *et al.*, 2018) directly, we construct dataset 3 by considering only chromosomes 15 to Y. We do not consider chromosomes 1 to 14 because the polyA site prediction model of DeepPASTA is trained on those chromosomes. We introduce dataset 3 in the performance evaluation because we want to show the performance of DeepPASTA not only on the PolyA-Seq data but also on the dataset from Conv-Net literature. We collect the positive sequences of dataset 3 from Leung *et al.* (2018). As in Leung *et al.* (2018), the negative sequences are constructed by shifting each positive example left and right by 50 nt. Therefore, the numbers of positive and negative sequences in dataset 3 are 6018 and 12 036, respectively. For each sequence, three energy efficient RNA secondary structures are constructed using RNAshapes.

As the Conv-Net model is not publicly available, we construct the model using the description in Leung *et al.* (2018) and train it using the sequences of the above training and validation data. We also train DeepPolyA using the same training and validation data because the tool was initially developed for plants. Since the tool Dragon PolyA Spotter needs sequences of length more than 200 nt as the input, we extend each sequence by 50 nt in both directions to make it 300 nt for Dragon PolyA Spotter. The performance of all the tools is compared using AUC and AUPRC.

From Table 1, it can be seen that DeepPASTA clearly outperforms the other tools in polyA site prediction. DeepPolyA and Conv-Net perform better than PolyAR, Dragon PolyA Spotter and DeeReCT-PolyA because they also use machine extracted features and deep learning algorithms. DeepPolyA preforms slightly better than Conv-Net perhaps because Conv-Net was originally designed to predict relatively dominant polyA sites. Dragon PolyA Spotter and DeeReCT-PolyA only predict hexamer signals in sequences, but

Table 1. Performance comparison between DeepPASTA, PolyAR, Dragon PolyA Spotter, DeeReCT-PolyA, Conv-Net and DeepPolyA in polyA site prediction on the three datasets introduced in the beginning of Section 3.1 in terms of AUC and AUPRC

Tool name	Performance Metric	Dataset 1	Dataset 2	Dataset 3
DeepPASTA	AUC	0.972	0.958	0.930
	AUPRC	0.921	0.962	0.875
PolyAR	AUC	0.630	0.713	0.673
	AUPRC	0.296	0.749	0.489
Dragon PolyA	AUC	0.609	0.711	0.639
Spotter	AUPRC	0.261	0.693	0.421
DeeReCT-PolyA	AUC	0.637	0.711	0.659
	AUPRC	0.261	0.695	0.421
Conv-Net	AUC	0.910	0.899	0.907[a]
	AUPRC	0.782	0.913	0.853
DeepPolyA	AUC	0.925	0.913	0.906
	AUPRC	0.804	0.922	0.854

[a]The AUC performance of Conv-Net on dataset 3 is taken from Leung *et al.* (2018).

a polyA site depends on other signals as well as the hexamer signal. As a result, Dragon PolyA Spotter and DeeReCT-PolyA perform the worst among the tools. Because PolyAR considers other signals along with the hexamer signal in its prediction process, it is able to perform better than Dragon PolyA Spotter and DeeReCT-PolyA. Moreover, all the tools generally perform better on dataset 2 than on dataset 1 as expected, but DeepPASTA drops slightly in AUC on dataset 2. In order to explain this, we conduct several experiments with different sets of negative examples as described above. We find that DeepPASTA performs the best on the shifted negative examples and it performs better on negative examples with hexamer signals than random negative examples (Supplementary Fig. S7). As dataset 2 does not contain shifted negative examples and negative examples with hexamer signals, its AUC drops slightly. On the dataset from the Conv-Net work (dataset 3), DeepPASTA clearly performs better than Conv-Net in both AUC and AUPRC. Hence, although both DeepPASTA, Conv-Net and DeepPolyA use deep learning algorithms, DeepPASTA performs significantly better than Conv-Net and DeepPolyA on all three datasets. The use of RNA secondary structures and recurrent neural networks in the model architecture may have helped improve the performance of DeepPASTA.

Two examples illustrating the contribution of RNA secondary structures are given in Supplementary Figure S8. Moreover, the hexamer signals that contributed to the performance of DeepPASTA in polyA site prediction are analyzed in Supplementary Figure S9.

3.1.1 Effect of data leak on polyA site prediction

In order to test the effect of data leak on our polyA site prediction method, we compare the performance of our DeepPASTA model trained in the above (where all homologous genes were consolidated in the training and validation data to prevent potential data leak in testing) with another one where the homologous genes are not consolidated. More specifically, we construct the training and validation data based on chromosomes similarly as above but we do not move homologous genes from dataset 1 to the training and validation data. As a result, the numbers of positive examples in the training and validation data are 251 726 and 125 665, respectively. Again, the negative examples are down-sampled to make the ratio of positive and negative examples 1 : 1 in this training and validation data. For convenience, let us refer to the DeepPASTA model trained with the new data as M2. The performance of the two models are compared using AUC and AUPRC on dataset 1 in Table 2. As shown in the table, there is only slight performance difference between the models. This negligible difference demonstrates that homology does not cause serious data leak in our polyA site prediction method.

3.1.2 Performance improvement using an RNN and RNA secondary structures

In order to test how the use of an RNN and RNA secondary structures may contribute to the performance of DeepPASTA, we consider three polyA site prediction models (see Supplementary Fig. S2): (i) the full polyA site prediction model of DeepPASTA, (ii) the model that uses only the sequence features (called M3) and (iii) the model that uses the sequence features and CNN (called M4). The models M3 and M4 are trained using the same training and validation data as in the training of full model of DeepPASTA. Note that these models have much less network complexity than DeepPASTA. The performance of the models is evaluated using AUC and AUPRC on datasets 1 and 2 in Table 3. The table shows that both RNN and RNA secondary structures make small (but non-negligible) contributions to the improved performance of DeepPASTA.

Table 2. The effect of data leak on DeepPASTA in polyA site prediction on dataset 1 in terms of AUC and AUPRC

PolyA site prediction model	AUC	AUPRC
DeepPASTA (homologs consolidated)	0.9724	0.9210
M2 (homologs not consolidated)	0.9725	0.9211

3.2 Performance on predicting tissue-specific polyA sites

Different human tissues can have different polyA sites (Tian and Manley, 2017) and the tissue specificity of polyA sites has been studied extensively in the literature (Hafez *et al.*, 2013; Tian and Manley, 2017; Weng *et al.*, 2016; Zhang *et al.*, 2005). In this subsection, we analyze the performance of the DeepPASTA (multi-label) model for tissue-specific polyA site prediction and compare it with that of the basic (i.e. non-tissue-specific) model and DeepPolyA (Gao *et al.*, 2018). Similar to the training of the basic model, the tissue-specific polyA site prediction model is also trained by consolidating homologous genes to prevent potential data leak. The training and validation data of this model are similar to the training and validation data of the basic model, but their ground truths are different. For each example in the training and validation data, the ground truth consists of nine labels (actually, score values) for nine tissues (brain, kidney, liver, maqc_brain1, maqc_brain2, maqc_UHR1, maqc_UHR2, muscle and testis), indicating if a tissue is likely to have a polyA site or not. Similar to the basic model, the hyperparameters of the tissue-specific model are tuned empirically using held-out validation data. The performance of the tissue-specific model is compared with DeepPolyA and the basic model on datasets 1 and 2 using AUC and AUPRC in Table 4. While evaluating the tissue-specific performance, if an input sequence contains a polyA site in the middle for a given tissue then the sequence is a positive example for that tissue, otherwise it's a negative example.

From the table, we can see that the tissue-specific model of DeepPASTA performs better than DeepPolyA and the non-tissue-specific model (of DeepPASTA) in predicting tissue-specific polyA sites. In fact, the two DeepPASTA models significantly outperform DeepPolyA in tissue-specific polyA site prediction. Although the AUC differences between the DeepPASTA models are very small, the improvements in AUPRC are quite significant. Clearly, the use of all nine tissues simultaneously in training of the tissue-specific model has helped its performance. Although, the AUCs of both models decrease from dataset 1 to dataset 2, their AUPRCs increase slightly. The reason of this performance variation between datasets 1 and 2 is due to the types of negative examples in the datasets

Table 3. Contributions of the RNN and RNA secondary structures in polyA site prediction on datasets 1 and 2 in terms of AUC and AUPRC

PolyA site prediction model	Dataset	AUC	AUPRC
DeepPASTA	Dataset 1	0.972	0.921
(sequences + RNA secondary structures)	Dataset 2	0.958	0.962
M3 (sequences with CNN and RNN)	Dataset 1	0.960	0.893
	Dataset 2	0.938	0.947
M4 (sequences with CNN)	Dataset 1	0.951	0.871
	Dataset 2	0.931	0.940

Table 4. Performance comparison between the tissue-specific model of DeepPASTA, DeepPolyA and basic (i.e. non-tissue-specific) polyA site prediction models of DeepPASTA on datasets 1 and 2

Tissue	Data	DeepPolyA		Tissue-specific		Basic model	
		AUC	AUPRC	AUC	AUPRC	AUC	AUPRC
Brain	D1	0.883	0.202	0.921	0.296	0.916	0.244
	D2	0.777	0.224	0.823	0.313	0.804	0.250
Kidney	D1	0.892	0.214	0.927	0.301	0.921	0.255
	D2	0.789	0.235	0.834	0.316	0.811	0.261
Liver	D1	0.878	0.170	0.916	0.251	0.910	0.207
	D2	0.767	0.189	0.813	0.265	0.792	0.213
MAQC_Brian1	D1	0.899	0.220	0.938	0.328	0.926	0.270
	D2	0.801	0.241	0.857	0.344	0.822	0.275
MAQC_Brian2	D1	0.898	0.207	0.937	0.314	0.924	0.258
	D2	0.799	0.227	0.855	0.329	0.820	0.263
MAQC_UHR1	D1	0.892	0.237	0.928	0.336	0.920	0.287
	D2	0.791	0.261	0.839	0.354	0.812	0.294
MAQC_UHR2	D1	0.892	0.250	0.928	0.348	0.921	0.299
	D2	0.792	0.276	0.840	0.367	0.813	0.306
Muscle	D1	0.877	0.207	0.910	0.272	0.912	0.251
	D2	0.765	0.232	0.806	0.302	0.792	0.258
Testis	D1	0.876	0.211	0.908	0.265	0.910	0.253
	D2	0.764	0.237	0.802	0.299	0.789	0.261

Note: Supplementary Table S1 shows the numbers of positive and negative examples in the test datasets. Datasets 1 and 2 are represented as D1 and D2, respectively, in the table.

(similar to the AUC variation observed in Section 3.1). Here, a sequence with the PASs may not have polyA sites in every tissue. This makes the prediction task much harder for the tissue-specific model of DeepPASTA on both datasets. Moreover, the AUPRC values are much lower than the AUC values because the negative examples greatly outnumber the positive examples on both datasets. Similar to the basic model, the hexamer signals that contributed to the performance of DeepPASTA in tissue-specific polyA site prediction are analyzed in Supplementary Figures S10–S18.

3.3 Performance on predicting tissue-specific relatively dominant polyA sites

In this subsection, we compare DeepPASTA with Conv-Net in predicting relatively dominant polyA sites in a particular tissue on two datasets: datasets 4 and 5. Dataset 4 contains nine tissues studied above (Derti *et al.*, 2012) and for each tissue, the human chromosomes are partitioned into three groups: chromosomes 1 to 6 as group 1, chromosomes 7 to 12 as group 2 and chromosomes 13 to Y as group 3. Again, homologs are consolidated to prevent potential data leak. All pairs of polyA sites of a particular gene from the PolyA-Seq data in a specific tissue are considered as examples (ordered by their genomic locations). We construct training, validation and test data from the examples in groups 1, 2 and 3, respectively. The tissue-specific read counts (from the PolyA-Seq data) are used to define the true relative dominance. For simplicity, we do not consider examples consisting of polyA sites with equal read counts. For each tissue, a model is trained and tested using the data of that tissue. Dataset 5 is taken from Leung *et al.* (2018). The training, validation and test parts of dataset 5 are constructed following the same construction steps as for dataset 4. Note that homologs are not consolidated in this dataset. There are eight tissues in this dataset and for each tissue, a model is again trained and tested using the data of that tissue. Again, the hyperparameters of the models are tuned empirically using held-out validation data. For each tissue of datasets 4 and 5, a Conv-Net model is trained with the same training and validation data as DeepPASTA. Supplementary Figure S19 shows the numbers of training and validation examples in datasets 4 and 5. The performance of DeepPASTA and Conv-Net in predicting tissue-specific relative dominance is compared using AUC and AUPRC in Table 5 and Supplementary Table S2. Clearly, DeepPASTA achieves a better overall performance and its improvement over Conv-Net is consistent across both datasets.

Table 5. Performance comparison between DeepPASTA and Conv-Net in relatively dominant polyA site prediction on dataset 4 in terms of AUC and AUPRC

Tissue	# of Test	DeepPASTA		Conv-Net	
	Examples	AUC	AUPRC	AUC	AUPRC
Brain	38 726	0.748	0.729	0.728	0.716
Kidney	44 363	0.708	0.694	0.699	0.679
Liver	39 832	0.713	0.698	0.676	0.664
MAQC_Brian1	44 242	0.723	0.707	0.714	0.693
MAQC_Brian2	40 878	0.709	0.694	0.690	0.673
MAQC_UHR1	62 064	0.704	0.704	0.689	0.696
MAQC_UHR2	62 946	0.721	0.707	0.691	0.682
Muscle	49 528	0.719	0.706	0.717	0.716
Testis	53 820	0.733	0.714	0.721	0.709

Note: The performance of Conv-Net is based on our implementation of the method described in Leung *et al.* (2018).

3.4 Performance on predicting tissue-specific absolutely dominant polyA sites

In this subsection, we compare the performance of DeepPASTA and Conv-Net in predicting absolutely dominant polyA sites in a particular tissue. Similar to dataset 4 constructed in a Section 3.3, the human chromosomes are partitioned and homologs are consolidated to construct a new dataset, called dataset 6. Among all the polyA sites of each gene, those that have the highest read counts in the

Table 6. Performance comparison between DeepPASTA and Conv-Net in predicting absolutely dominant polyA sites on dataset 6 in terms of AUC and AUPRC

Tissue	Pos #	Neg #	DeepPASTA		Conv-Net	
			AUC	AUPRC	AUC	AUPRC
Brain	5420	11 859	0.703	0.533	0.641	0.475
Kidney	5362	12 964	0.698	0.508	0.647	0.452
Liver	5006	11 911	0.688	0.496	0.651	0.465
MAQC_Brain1	5365	13 015	0.720	0.517	0.625	0.415
MAQC_Brain2	5256	12 196	0.723	0.531	0.650	0.443
MAQC_UHR1	5633	16 018	0.699	0.466	0.640	0.390
MAQC_UHR2	5762	16 885	0.712	0.476	0.659	0.424
Muscle	5545	14 240	0.693	0.488	0.643	0.420
Testis	5553	14 669	0.693	0.485	0.649	0.428

Note: The second and third columns give the number of positive and negative examples in the test data.

PolyA-Seq data with respect to a particular tissue are considered as the absolutely dominant polyA sites of the gene (in the tissue). The rest of the polyA sites are considered as non-dominant polyA sites of the gene (i.e. negative examples) in the tissue. Supplementary Figure S20 shows the numbers of training and validation examples in dataset 6. The performance of DeepPASTA in predicting absolutely dominant polyA sites is compared with Conv-Net using AUC and AUPRC on the test data of dataset 6, as shown in Table 6. DeepPASTA clearly outperforms Conv-Net in all tissues. This significant performance improvement of DeepPASTA can be partially attributed to its use of RNA secondary structures.

4 Discussion

In this work, we introduced DeepPASTA, a deep learning-based tool for predicting polyA sites from genomic sequence and RNA secondary structure data. The tool is also capable of predicting tissue-specific polyA sites as well as tissue-specific relatively and absolutely dominant polyA sites. Our extensive experiments show that DeepPASTA performs better than all existing tools in all four polyA site analyses. Supplementary Table S3 illustrates that the four polyA site prediction models of DeepPASTA can be trained in a reasonable amount of time. Hence, we expect that DeepPASTA will serve as a useful polyA site analysis tool in biological research.

Funding

This work was supported in part by NSF grant IIS-1646333, NIH grant U01HG009417 and NSFC grant 61370172.

Conflict of Interest: none declared.

References

Akhtar,M.N. *et al.* (2010) Polyar, a new computer program for prediction of poly(A) sites in human sequences. *BMC Genomics*, **11**, 646.

Alipanahi,B. *et al.* (2015) Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, **33**, 831–838.

Aly,M. (2005) Survey on multiclass classification methods. *Technical report*. California Institute of Technology.

Angermueller,C. *et al.* (2017) DeepCpG: accurate prediction of single-cell DNA methylation states using deep learning. *Genome Biol.*, **18**, 67.

Bajic,B. *et al.* (2012) Dragon PolyA Spotter: prediction of poly(A) motifs within human genomic sequences. *Bioinformatics*, **28**, 127–129.

Bar-Shira,A. *et al.* (1991) An RNA secondary structure juxtaposes two remote genetic signals for human T-cell leukemia virus type I RNA 3′-end processing. *J. Virol.*, **65**, 5165–5173.

Barrett,L.W. *et al.* (2012) Regulation of eukaryotic gene expression by the untranslated gene regions and other non-coding elements. *Cell. Mol. Life Sci.*, **69**, 3613–3634.

Bishop,C.M. (2006) *Pattern Recognition and Machine Learning*. Springer.

Brown,P.H. *et al.* (1991) Effect of RNA secondary structure on polyadenylation site selection. *Genes Dev.*, **5**, 1277–1284.

Cheng,Y. *et al.* (2006) Prediction of mRNA polyadenylation sites by support vector machine. *Bioinformatics*, **22**, 2320–2325.

Ciregan,D.F. *et al.* (2012) Multi-column deep neural networks for image classification. In: *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA.

Colgan,D.F. *et al.* (1997) Mechanism and regulation of mRNA polyadenylation. *Genes Dev.*, **11**, 2755–2766.

Danckwardt,S. *et al.* (2008) 3′ end mRNA processing: molecular mechanisms and implications for health and disease. *EMBO J.*, **27**, 482–498.

Derti,A. *et al.* (2012) A quantitative atlas of polyadenylation in five mammals. *Genome Res.*, **22**, 1173–1183.

Di Giammartino, D. *et al.* (2011) Mechanisms and consequences of alternative polyadenylation. *Mol. Cell*, **43**, 853–866.

Dozat,T. (2016) *Incorporating Nesterov Momentum into Adam*. ICLR Workshop Paper. Caribe Hilton, San Juan, Puerto Rico.

Eser,U. and Churchman,L.S. (2016) FIDDLE: an integrative deep learning framework for functional genomic data inference. doi: 10.1101/081380.

Gao,X. *et al.* (2018) DeepPolyA: a convolutional neural network approach for polyadenylation site prediction. *IEEE Access*, **6**, 24340–24349.

Gers,F.A. *et al.* (2000) Learning to forget: continual prediction with LSTM. *Neural Comput.*, **12**, 2451–2471.

Hafez,D. *et al.* (2013) Genome-wide identification and predictive modeling of tissue-specific alternative polyadenylation. *Bioinformatics*, **29**, 108–116.

He,K. *et al.* (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: *International Conference on Computer Vision*, pp. 1026–1034.

Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.

Hu,J. (2005) Bioinformatic identification of candidate cis-regulatory elements involved in human mRNA polyadenylation. *RNA*, **11**, 1485–1493.

Kelley,D.R. *et al.* (2016) Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.*, **26**, 990–999.

Lange,S.J. *et al.* (2012) Global or local? Predicting secondary structure and accessibility in mRNAs. *Nucleic Acids Res.*, **40**, 5215–5226.

LeCun,Y. *et al.* (1998) Gradient-based learning applied to document recognition. *Proc. IEEE*, **86**, 2278–2323.

Leung,M.K.K. *et al.* (2018) Inference of the human polyadenylation code. *Bioinformatics*, **34**, 2889–2898.

Lin,Y. *et al.* (2012) An in-depth map of polyadenylation sites in cancer. *Nucleic Acids Res.*, **40**, 8460–8471.

Liu,H. *et al.* (2003) An in-silico Method for Prediction of Polyadenylation Signals in Human Sequences. *Genome Inf.*, **14**, 84–93.

Liu,H. *et al.* (2005) DNAFSMiner: a web-based software toolbox to recognize two types of functional sites in DNA sequences. *Bioinformatics*, **21**, 671–673.

Maticzka,D. *et al.* (2014) GraphProt: modeling binding preferences of RNA-binding proteins. *Genome Biol.*, **15**, R17.

Mayr,C. (2016) Evolution and Biological Roles of Alternative 3′ UTRs. *Trends Cell Biol.*, **26**, 227–237.

Nair,V. and Hinton,G.E. (2010) Rectified linear units improve restricted Boltzmann machines. In: *International Conference on Machine Learning*. Haifa, Israel.

Pan,X. and Shen,H.B. (2017) RNA-protein binding motifs mining with a new hybrid deep learning based cross-domain knowledge integration approach. *BMC Bioinformatics*, **18**, 136.

Pichon,X. *et al.* (2012) RNA binding protein/RNA element interactions and the control of translation. *Curr. Protein Pept. Sci.*, **13**, 294–304.

Salamov,A.A. *et al.* (1997) Recognition of 3′ -processing sites of human mRNA precursors. *Bioinformatics*, **13**, 23–28.

Schuster,M. and Paliwal,K.K. (1997) Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.*, **45**, 2673–2681.

Shaw,G. and Kamen,R. (1986) A conserved AU sequence from the 3′ untranslated region of GM-CSF mRNA mediates selective mRNA degradation. *Cell*, **46**, 659–667.

Srivastava,N. *et al.* (2014) Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.

Steffen,P. *et al.* (2006) RNAshapes: an integrated RNA analysis package based on abstract shapes. *Bioinformatics*, **22**, 500–503.

Tabaska,J.E. *et al.* (1999) Detection of polyadenylation signals in human DNA sequences. *Gene*, **231**, 77–86.

Tian,B. *et al.* (2005) A large-scale analysis of mRNA polyadenylation of human and mouse genes. *Nucleic Acid Res.*, **33**, 201–212.

Tian,B. and Manley,J.L. (2017) Alternative polyadenylation of mRNA precursors. *Nat. Rev. Mol. Cell Biol.*, **18**, 18–30.

Wahle,E. and Kühn,U. (1997) The mechanism of 3′ cleavage and polyadenylation of eukaryotic pre-mRNA. *Nucleic Acid Res. Mol. Biol.*, **57**, 41–71.

Wahle,E. *et al.* (1995) 3′ End cleavage and polyadenylation of mRNA precursors. *Biochim. Biophys. Act*, **1261**, 183–194.

Weng,L. *et al.* (2016) Poly(A) code analyses reveal key determinants for tissue-specific mRNA alternative polyadenylation. *RNA*, **22**, 813–821.

Wu,C. *et al.* (2004) Secondary structure as a functional feature in the downstream region of mammalian polyadenylation signals. *Mol. Cell. Biol.*, **24**, 2789–2796.

Xia,Z. *et al.* (2018) DeeReCT-PolyA: a robust and generic deep learning method for PAS identification. *Bioinformatics*, dbioinformatics/bioinformatics/bty991.

Yada,T. *et al.* (1994) Statistical analysis of human DNA sequences in the vicinity of poly(A) signal. ICOT Technical Report TR-876.

Zhang,H. *et al.* (2005) Biased alternative polyadenylation in human tissues. *Genome Biol.*, **6**, R100.

Zhang,S. *et al.* (2015) A deep learning framework for modeling structural features of RNA-binding protein targets. *Nucleic Acids Res.*, **44**, e32.

Zhang,S. *et al.* (2017a) TITER: predicting translation initiation sites by deep learning. *Bioinformatics*, **33**, 234–242.

Zhang,S. *et al.* (2017b) ROSE: a deep learning based framework for predicting ribosome stalling. *Res. Comput. Mol. Biol.*, **21**, 402–403.

Zhou,J. *et al.* (2015) Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods*, **12**, 931.