Sequence analysis

SANPolyA: a deep learning method for identifying Poly(A) signals

Haitao Yu [1], Zhiming Dai [1, 2,*]

[1] School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 510006, China

[2] Guangdong Province Key Laboratory of Big Data Analysis and Processing, Sun Yat-Sen University, Guangzhou 510006, China

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Polyadenylation plays a regulatory role in transcription. The recognition of polyadenylation signal (PAS) motif sequence is an important step in polyadenylation. In the past few years, some statistical machine learning-based and deep learning-based methods have been proposed for PAS identification. Although these methods predict PAS with success, there is room for their improvement on PAS identification.

Results: In this study, we proposed a deep neural network-based computational method, called SANPolyA, for identifying PAS in human and mouse genomes. SANPolyA requires no manually crafted sequence features. We compared our method SANPolyA with several previous PAS identification methods on several PAS benchmark datasets. Our results showed that SANPolyA outperforms the state-of-art methods. SANPolyA also showed good performance on leave-one-motif-out evaluation.

Availability: https://github.com/yuht4/SANPolyA

Contact: daizhim@mail.sysu.edu.cn

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

1 Introduction

Polyadenylation is an important cellular process in transcription, which involves in the production of mature messenger RNA (mRNA). The polyadenylation mechanism is quite concise. Polyadenylation triggers upon the transcription termination of a gene. The 3'-most segment of the precursor mRNA (pre-mRNA), newly produced, is cleaved off first, then a poly(A) tail is added at the RNA's 3'-end (Liu, et al., 2003). The polyadenylation process is catalyzed by a multi-protein polyadenylation complex, which contains the cleavage and polyadenylation specific factors (CPSF), cleavage stimulation factors (CstF), polyadenylate polymerases (PAP), polyadenylate binding proteins (PAB) and cleavage factors (Hunt, et al., 2008). The polyadenylation signal (PAS) sequence is important for the binding and function of the multi-protein polyadenylation complex. The PAS sequence is around 10~30 nucleotides upstream of the RNA cleavage site (Proudfoot, 2011). The PAS sequence is considered as the sequence motif, recognized by the multi-protein RNA cleavage complex. The sequence composition of PAS motif varies among different species. For example, in human, most polyadenylation sites contain the AAUAAA sequence (Beaudoing, et al., 2000), while such sequence is not very common in fungi and plants (Shen, et al., 2008). Moreover, even in the same species, the PAS motifs also vary. For example, Tian, et al, found that a large proportion (~50%) of human genes have alternative PAS motifs (Tian, et al., 2005).

The flanking sequence elements of PAS motif are also important for the cleavage and polyadenylation. For example, downstream region of the RNA cleavage site has GU-rich sequence elements. These GU-rich sequences are bound by the cleavage stimulation factor (CstF) of the multi-protein RNA cleavage complex (Beaudoing, et al., 2000; Proudfoot, 2011). The upstream and downstream region of the PAS motif contains U-rich sequence elements (Tian, et al., 2005). The upstream and downstream regions around PAS motif can regulate polyadenylation efficiency (Chen, et al., 1995; Proudfoot, 2011). It was reported that the upstream and downstream sequences can be considered as useful sequence patterns to recognize the PAS motif-contained sequences (Zarudnaya, et al., 2003). However, unlike the PAS motif, upstream and downstream sequence elements around the PAS motif have

quite high variability, which is a challenge in identifying PAS from genome sequence (Salamov and Solovyev, 1997). The accurate identification of PAS can help us understand regulatory mechanisms of polyadenylation process and contribute to better gene annotation. It was reported that mutations of PAS motifs and their surrounding sequence elements are associated with diseases (Lin, et al., 2012).

In the past years, based on statistical information of PAS surrounding sequence and many hand-crafted DNA features, researchers have proposed many methods for analyzing and predicting PAS in mRNA and DNA sequences (Akhtar, et al., 2010; Hu, et al., 2005; Liu, et al., 2003; Matis, et al., 1996; Miura, et al., 2006; Salamov and Solovyev, 1997; Tabaska and Zhang, 1999; van Helden, et al., 2000). The sequences features surrounding the PAS motifs were used to discriminate the true PAS with the pseudo one. The pseudo PAS sequence contains the same PAS motif, as the true PAS sequence does. However, the pseudo PAS sequence does not link to the RNA polyadenylation event. In recent years, researchers have made great progress in identifying Poly(A) signals in human genome and have proposed many state-of-art methods. Kalkatawi et al. developed a tool, called Dragon PolyA Spotter (DPA in abbreviation), based on properties of DNA sequences surrounding PAS motifs, including statistical, thermodynamic and physico-chemical characteristics. This tool outperformed previous tools in predicting 12 common PAS motifs in human genome (Kalkatawi, et al., 2012). Moreover, Kalkatawi et al. processed and collected a dataset (denoted as dragon-human) for 12 most common PAS motif variants in human genome. Xie, et al, proposed a machine learning-based method for PAS identification by combining support vector machine and hidden Markov model (Xie, et al., 2013). We denoted this tool as HMM-SVM for convenience. It was reported that HMM-SVM got a higher accuray on the dragon-human dataset than DPA (Xie, et al., 2013). Recently, Magana-Mora, et al. developed a tool, called Omni-PolyA, by combining several machine learning methods (Magana-Mora, et al., 2017). It was reported that Omni-PolyA outperformed HMM-SVM on the dragon-human dataset for predicting the 12 most common PAS variants. Magana-Mora, et al, proposed another benchmark dataset (denoted as omni-human) for the 12 most common PAS variants in human genome. DPA, HMM-SVM and Omni-PolyA need to train a specific model for every human PAS motif variant. DPA and Omni-PolyA are feature-based, which require the DNA sequence features surrounding PAS motifs. Recently, some deep-learning based methods have been proposed for PAS identification (Albalawi, et al., 2019; Arefeen, et al., 2019; Kalkatawi, et al., 2019; Leung, et al., 2018; Xia, et al., 2018). Xia, et al, proposed a deep learning-based tool for PAS identification, DeeReCT-PolyA, which needs no prior knowledge of DNA sequence features (Xia, et al., 2018). DeeReCT-PolyA used only a single generic model to identify 12 common PAS variants in human. DeeReCT-PolyA outperfromed previous tools on dragon-human dataset and omni-human dataset. Moreover, Xia, et al, collected two mouse datasets, C57BL/6J (denoted as BL) and SPRET/EiJ (denoted as SP), to evaluate the performance of DeeReCT-PolyA on different speices. Kalkatawi, et al, proposed a tool DeepGSR, which can be used to identify translation initiation sites and PAS (Kalkatawi, et al., 2019). DeepGSR was reported to outperform previous state-of-art tools in predicting AATAAA PAS motif variant in human genome. Moreover, Kalkatawi, et al, collected several PAS datasets for 16 common PAS variants in human, mouse, bovine and fruit fly. We denoted this human dataset as GSR-human. Albalawi, et al, proposed a hybrid model-based tool, called HybPAS, which contains 8 neural networks and 4 logistic regression models for PAS identification in human genome (Albalawi, et al., 2019). HybPAS constructed a specific model for each human PAS motif variant, a total of 12 models. However, there is room for improvement for the prediction of PAS signals in human and mouse genomes.

In this paper, we proposed a self-attention deep neural network-based computational tool, called SANPolyA, for identifying Poly(A) signals in human and mouse genomes. SANPolyA does not need to train specific model for every PAS motif. Instead, SANPolyA is a generic model that can identify most common PAS motifs. SANPolyA does not need any manually crafted sequence features and can learn high level abstract features. We trained and tested SANPolyA on several benchmark PAS datasets in human and mouse genomes. We found that SANPolyA performs better than previous state-of-art tools in prediction Poly(A) signals in human and mouse genomes. We conducted a leave-one-motif-out evaluation, and found that SANPolyA can predict the PAS motif variants that are not included in the training data.

2 Methods

2.1 Dataset

In this study, we used several previously collected poly(A) datasets, including dragon-human (Kalkatawi, et al., 2012), omni-human (Magana-Mora, et al., 2017), GSR-human (Kalkatawi, et al., 2019), C57BL/6J (BL) (Xia, et al., 2018), and SPRET/EiJ (SP) (Xia, et al., 2018). The samples in the PAS datasets are all PAS-like sequences. A PAS-like sequence is a DNA sequence that has the 6 bp long PAS motif in the middle. The true PAS sequences are the PAS-like sequences that link to the polyadenylation. The pseudo PAS sequences in our datasets are PAS-like sequences with the same PAS motifs as true PAS sequences do. However, pseudo-PAS sequences do not link to the polyadenylation. For each PAS motif variant, the number of pseudo PAS sequences is the same as that of the true PAS sequences.

Both dragon-human and omni-human datasets cover 12 most common PAS motif variants in human. The dragon-human dataset contains totally 14,740 PAS-like sequences samples, including 7370 true PAS sequences and 7370 pseudo PAS sequences. The omni-human dataset contains 18,786 true PAS sequences and the same number of pseudo PAS sequences. The GSR-human dataset contains 20,933 true PAS sequences and the same number pseudo PAS sequences for 16 most common PAS variants in human.

The C57BL/6J (BL) and SPRET/EiJ (SP) datasets cover 13 PAS motif variants in mouse, slightly different from the motif variants in human datasets. The dataset C57BL/6J (BL) contains 23,112 true PAS sequences and 23,112 pseudo PAS sequences. The dataset SPRET/EiJ (SP) contains 20,115 true PAS sequences and 20,115 pseudo PAS sequences.

The positive and negative samples in dragon-human, omni-human, C57BL/6J (BL), SPRET/EiJ (SP) and GSR-human datasets are all balanced, which means the ratio between the numbers of positive and negative samples is 1:1. In real condition, the number of pseudo PAS sequences may be larger than the number of positive sequences. We sought to evaluate our model performance on an unbalanced PAS dataset. We used the GENCODE poly(A) annotation data (Release 31) (Harrow, et al., 2012) and the human reference genome hg38 (Release 31) to get the true PAS from the human genome. For every true PAS, we extracted the sequence that is 100 bp upstream and 100 bp downstream around the 6bp long PAS motif. There are more than 600 kinds of the PAS motifs in the annotation data, and most kinds of the PAS motifs are very rare and have insufficient positive examples. We only considered a PAS variant if its number in the annotation data is more than 50. After filtering the repeated sequences, we got 38,527 true PAS sequences for 18 common PAS motif variants in human genome (Table S1). We

obtained the pseudo PAS sequences by randomly sampling in the genome, excluding the regions that cover 1000 bp upstream and 1000 bp downstream of the true PAS sequences. For each PAS motif variant, the ratios between the numbers of the pseudo and the true PAS sequences are 1:1, 2:1, and 4:1. We referred to our newly collected dataset as SAN-human.

2.2 The SANPolyA method

We built a deep neural network model for PAS identification, called SANPolyA, with self-attention mechanism.

2.2.1 Attention mechanism

In our method SANPolyA, we used the attention mechanism to capture the dependency relationship between different parts of the DNA sequences. The attention mechanism can make the model focus on some important sequence elements adaptively and learn the global and local information of the DNA sequences simultaneously.

In Figure 1a, we showed the mechanism of the scaled dot-product attention. The $Q \in \mathbb{R}^{n \times d_k}, K \in \mathbb{R}^{m \times d_k}, V \in \mathbb{R}^{m \times d_v}$ strands for three matrices: the queries, the keys, and the values. The scaled dot-product attention output is calculated as

$$\text{Attention}(Q,K,V) = \text{soft max}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The scaled dot-product attention operation can be viewed as a simple function, which changed the input $n \times d_k$ Q matrix to an output matrix $n \times d_v$. In Figure 1b, we showed the mechanism of the multi-head attention. In the task of PAS identification, Q, K and V are all the same and are denoted as the input tensor of the multi-head attention layer $X \in \mathbb{R}^{n \times d_b}$. The multi-head attention operation just repeats the scaled dot-product operation N times (N heads). For a single head, the scaled dot-product attention output was defined as below: $\text{head}_i = \text{Attention}\left(XW_i^{(1)}, XW_i^{(2)}, XW_i^{(3)}\right)$, where the trained parameters $W_i^{(1)} \in \mathbb{R}^{d_b \times d_o}$, $W_i^{(2)} \in \mathbb{R}^{d_b \times d_o}$, $W_i^{(3)} \in \mathbb{R}^{d_b \times d_o}$. Then the multi-head attention output is expressed as $\text{linear}(\text{Concat}(\text{head}_1,...,\text{head}_N))$, where Linear means Linear transformation, Concat means matrix concatenation. The parameters of $W_i^{(1)}$, $W_i^{(2)}$ and $W_i^{(3)}$ are different in different "heads".

Figure 1. The diagram of the attention mechanism we used. (a) The scaled dot-product attention mechanism (b) The multi-head attention mechanism.

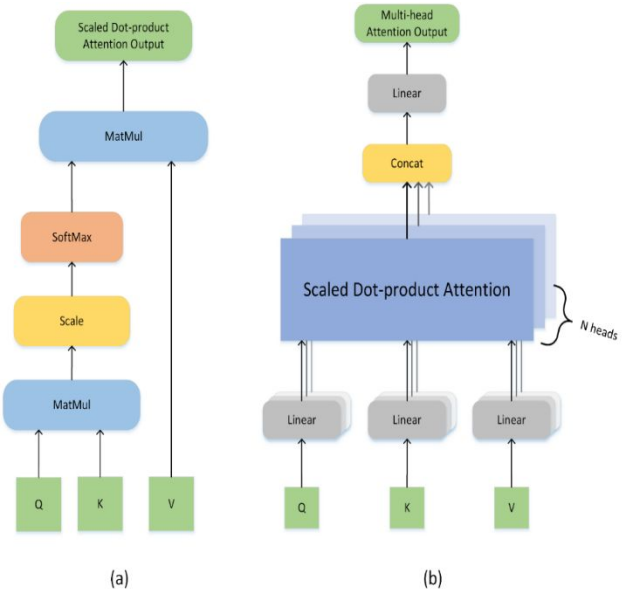
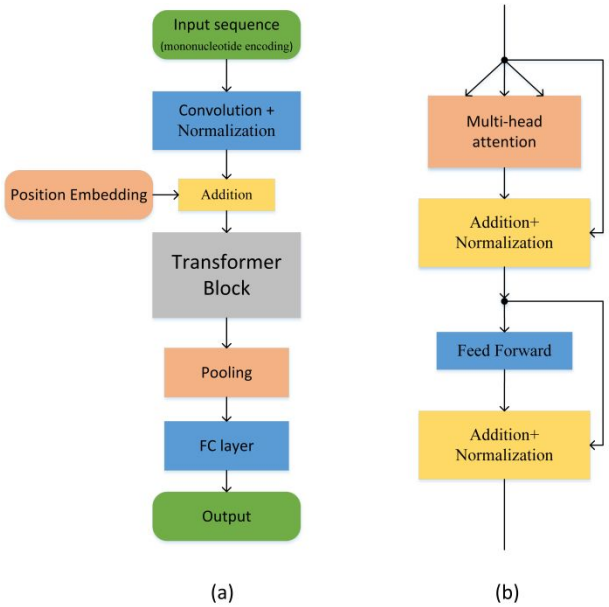


Figure 2. The architecture of our model we used. (a) Our proposed method SANPolyA. (b) Transformer block.



2.2.2 SANPolyA method

We showed the architecture of our SANPolyA method in Figure 2. The input of SANPolyA, is a PAS-like genomic DNA sequence encoded by the mononucleotide encoding method. The first component of SANPolyA is a one-dimensional convolution (Conv) layer with a group normalization (GN) layer (Wu and He, 2018). The Conv layer has 16 filters, with kernel size set as 10. We set the number of groups as 4 in the GN layer. The activation function in the first component is the Exponential Linear Unit (Clevert, et al., 2015). As we did not used the recurrent unit architecture in our method, we should account for the order of sequences in the input tensor. To solve the problem, we applied a common position embedding operation (Vaswani, et al., 2017) on the output tensor of the first component of SANPolyA. For convenience, we denoted the input tensor of the position embedding operation as $T \in \mathbb{R}^{n \times d_{in}}$, where d_{in} is the dimension of the input tensor. The Position Embedding method we used is as below:

$$PE_{2i}(p) = sin\left(\frac{p}{10000^{\frac{2i}{d_{pos}}}}\right), PE_{2i+1}(p) = cos\left(\frac{p}{10000^{\frac{2i}{d_{pos}}}}\right)$$

,where p is the position of T, $d_{pos}$is the dimension of positional encoding. The dimension of the positional encoding d_{pos} is the same as the dimension of the input T. For example, the 5th element (denoted as $T_{[5]}$) of the input T is a vector with dimension d_{in}. Then the ith element of the position embedding of the vector $T_{[5]}$ is calculated by $PE_i(5)$. After position embedding operation, we got the position information of the output tensor of the first component (Conv+GN) of SANPolyA. Then we added the position information to the original output of the first component. The following component of SANPolyA is a transformer block, we used the same transformer architecture as a previous study (Figure 2b) (Vaswani, et al., 2017). Transformer block contains a multi-head attention operation described in Section 2.2.1. We set the number of heads in the multi-head attention operation as 2 and the output dimension of each head as 8. The "Addition" in the transformer block means the element-wise addition and the "Normalization" in the transformer block

means the layer normalization (Lei Ba, et al., 2016). The feed forward layer in the transformer block is a naïve network with two dense layer, and each layer has 16 units.

Following the transformer block, we used a one-dimensional average pooling layer. We set the pooling window as 10. After the average pooling layer, we added a dropout layer to prevent overfitting. After dropout layer, we used a flatten layer to flatten the intermediate features. We set the number of hidden units in fully connected (FC) layer as 64. We used the Exponential Linear Unit as the activation function of FC layer. Finally, we used a sigmoid function to calculate the probability to evaluate whether the input PAS-like DNA sequence is true PAS sequence. Besides the dropout method, we used the L2 regularization for the weights and bias in all convolution layers and fully connected layers. As for optimizer and loss function, we decided to use stochastic gradient descent (SGD) with momentum and the binary cross-entropy. We set the training epoch as 500 and the mini-batch size as 64. We used the early stopping technique in training. The training will stop when the loss stops decreasing on the validation set after 40 training epochs. We saved the model that has highest accuracy on the validation set. During the training, the learning rate will be divided by 2 when the loss stopped decreasing on validation set after 25 epochs. In our study, we used five folds cross-validation strategy to evaluate the performance of our method, in order to be consistent with a previous method DeeReCT-PolyA (Xia, et al., 2018). In this study, we used this strategy unless other stated. We partitioned each dataset into five equally size folds respectively. In each iteration, we used three folds as training set, one fold as validating set, and the remaining one fold as testing set. In the training process, we pooled the data of all PAS motif variants in the training set together, and trained a single model. SANPolyA is a single generic deep learning model for all common PAS motif variants in the dataset. We kept the specific model that got the highest prediction accuracy on the validation set and we evaluated the performance of this model on the testing set. For each PAS motif variant, we calculated the average performance of SANPolyA over all iterations.

The hyper-parameters tuning strategies in DPA, HMM-SVM and Omni-PolyA are based on grid search, and the hyper-parameters tuning strategy in DeeReCT-PolyA, which has more hyper-parameters, is based on random sampling. Computational cost of grid search become very large with the increased number of hyper-parameters. Considering that the number of hyper-parameters in our method is comparable with that in DeeReCT-PolyA, we used random sampling to tune hyper-parameters with much less computational cost than grid search. In this study, we used two different hyper-parameters tuning strategies, random sampling strategy and manually searching strategy. When comparing the performance of our method with other methods, hyper-parameters tuning based on random sampling strategy was used.

Random sampling strategy: we randomly sampled the hyper-parameters during training (Table S2) and chose the set of parameters that achieved the best performance on the validation datasets for each dataset by five-fold-cross-validation, respectively (see parameters configuration for each dataset in Table S3). This hyper-parameters tuning strategy is the same as that in DeeReCT-PolyA.

Manually searching strategy: we determined the hyper-parameters empirically and fixed this configuration to keep the structure of our method simple and consistent. The hyper-parameters configuration was empirically determined as follows: dropout rate as 0.25, all L2 regularization rates as 0.0001, momentum rate as 0.95 and learning rate as 0.005.

We implemented our model SANPolyA via Keras. We used the default hyper-parameters in Keras (Chollet and others, 2015), except those that were mentioned explicitly in this paper.

2.2.3 Data representation

We considered four DNA sequences representation methods, including mononucleotide, dinucleotides, trinucleotides and embedding (see details in Supplementary materials). To decide the encoding strategy for our experiment, we evaluated the four encoding methods on dragon-human dataset (Figure S1). We found that mononucleotide representation method achieved better performance than the other methods. In addition, one advantage of mononucleotide method is that it is simpler and needs less computational cost than the other methods. We thus decided to use the mononucleotide representation method. Based on mononucleotide encoding method, A, T, C and G were encoded as the tuples: (1,0,0,0), (0,1,0,0), (0,0,1,0) and (0,0,0,1), respectively.

2.3 Method comparison strategy

We compared our method SANPolyA with several previous state-of-art tools, including DPA (Kalkatawi, et al., 2012), HMM-SVM (Xie, et al., 2013), DeeReCT-PolyA (Xia, et al., 2018), DeepGSR (Kalkatawi, et al., 2019) and Omni-PolyA (Magana-Mora, et al., 2017). In order to be consistent with comparison strategy used in previous studies (Magana-Mora, et al., 2017; Xia, et al., 2018), we used error rate to evaluate PAS identification performance. The error rate is defined as below:

$$error\_rate = 1 - \frac{TP + TN}{TP + TN + FP + FN},$$

where TP, TN, FP and FN mean true positive, true negative, false positive and false negative.

3 Results

We evaluated the performance of SANPolyA on the dragon-human, omni-human, C57BL/6J (BL), SPRET/EiJ (SP), GSR-human and our newly collcected SAN-human PAS datasets. We compared our method with two different deep learning architectures and found that our transformer block and self-attention mechanism can help improve prediction performance. We compared SANPolyA with the state-of-the-art PAS prediction tools. Results showed that SANPolyA performed better than the existing tools. We conducted a leave-one-motif-out evaluation on the omni-human dataset, and showed that SANPolyA also performed well. In addition, we visualized the sequence motifs learned by SANPolyA.

3.1 Comparison with other deep learning architectures

Transformer block in SANPolyA contains self-attention mechanism (e.g. multi-head attention operation) which can capture the relational and sequential information of different parts in a tensor. On the other hand, long short-term memory (LSTM) involves sequential manipulation, which cannot easily be run in parallel. Transformer block can be trained much faster than LSTM because transformer block only contains matrix operation. To verify the benefit of the transformer block used in SANPolyA, we compared SANPolyA with two other deep learning architectures: pure convolutional neural network (CNN) based model and LSTM based model. We got the pure CNN model by removing the transformer block and the position embedding operation in SANPolyA. We denoted the pure CNN model as SANPolyA-pureCNN. We got the LSTM based model by removing transformer block and position

embedding operation, and adding a bidirectional LSTM layer in the original position of the transformer block in SANPolyA. We set the number of units in the LSTM layer as 8, in order to keep the output dimension of the bidirectional LSTM layer in SANPolyA-LSTM the same as that of the transformer block in SANPolyA, which can make the number of parameters similar between SANPolyA-LSTM and SANPolyA. We denoted the LSTM based model as SANPolyA-LSTM.

We compared the performance of these three models on the dragon-human dataset. We calculated the weighted average error rate. The weight of every PAS variant is the sample number of the variant. Both SANPolyA and SANPolyA-LSTM showed performance improvement compared with SANPolyA-pureCNN (Table 1). The improvement of SANPolyA is more significant than that of SANPolyA-LSTM. We also compared the total time of the training and testing of these models (Table 1). The GPU version we used in this paper is Tesla V100-SXM2-16GB. The time consuming of the LSTM based model is much larger than the original SANPolyA and SANPolyA-pureCNN. These results showed that our transformer block can help improve PAS identification better than the LSTM module in terms of prediction performance and time consuming.

Table 1. Comparison performance of three models on dragon-human dataset.

	SANPolyA	SANPolyA -pureCNN	SANPolyA -LSTM
Weighted error rate	6.73%	9.21%	8.30%
Time consuming	247.00s	219.94s	22807.65s

3.2 Performance on PAS datasets

We compared our SANPolyA with previous PAS signal prediction tools, including DPA (Kalkatawi, et al., 2012), HMM-SVM (Xie, et al., 2013), DeeReCT-PolyA (Xia, et al., 2018), and Omni-PolyA (Magana-Mora, et al., 2017). We conducted the comparison on the dragon-human, omni-human, C57BL/6J (BL), and SPRET/EiJ (SP) datasets, respectively. The performance results of DPA, HMM-SVM, Omni-PolyA and DeeReCT-PolyA were quoted from previous studies (Magana-Mora, et al., 2017; Xia, et al., 2018). SANPolyA outperformed DPA, HMM-SVM, Omni-PolyA and DeeReCT-PolyA in most motifs on dragon-human and omni-human datasets (Table 2, 3, S4, S5).

As for mouse datasets, C57BL/6J (BL), and SPRET/EiJ (SP), we compared our SANPolyA with DeeReCT-PolyA which was also built for mouse PAS identification based on the two mouse datasets (Xia, et al., 2018). DPA, HMM-SVM and Omni-PolyA were built for human PAS identification, which cannot be applied for mouse directly. Compared with DeeReCT-PolyA, SANPolyA got an improvement on error rate for identification of each PAS motif variant (Table 4, S6, S7). The performance results of DeeReCT-PolyA on SPRET/EiJ (SP) and C57BL/6J (BL) datasets were quoted from the original study (Xia, et al., 2018).

Table 2 Error rates comparison between SANPolyA and previous tools on the dragon-human dataset. The best performance for each PAS motif variant was indicated in bold.

Variants	Size	Error Rate (%)				
		DPA	HMM -SVM	Omni-PolyA	DeeReCT-PolyA	SANPolyA
AATAAA	5190	23.72	28.13	14.02	11.81	**8.69**
ATTAAA	2400	16.63	23.96	12.50	9.00	**7.83**
AAGAAA	1250	14.00	10.96	10.80	7.76	**6.40**
AAAAAG	1230	8.05	8.62	4.87	5.77	**4.15**
AATACA	880	20.00	19.89	13.52	10.45	**6.14**
TATAAA	780	18.08	16.79	13.85	7.69	**6.67**
ACTAAA	690	23.33	26.38	14.49	10.72	**7.54**
AGTAAA	670	19.55	23.13	13.13	9.55	**6.27**
GATAAA	460	21.74	12.83	8.48	8.04	**5.87**
AATATA	410	18.05	14.15	13.41	8.78	**7.32**
CATAAA	410	20.00	14.15	14.39	9.02	**7.56**
AATAGA	370	18.38	8.11	11.62	**4.59**	5.14

Table 3 Error rates comparison between SANPolyA and previous tools on the omni-human dataset. The best performance for each PAS motif variant was indicated in bold.

Variants	Size	Error Rate (%)				
		DPA	HMM-SVM	Omni-PolyA	DeeReCT-PolyA	SANPolyA
AATAAA	24,310	25.49	27.91	23.96	21.99	**13.99**
ATTAAA	7098	25.59	33.48	24.20	23.01	**14.14**
TATAAA	1640	26.52	36.83	25.86	23.60	**15.67**
AGTAAA	1306	26.67	34.77	23.07	20.21	**13.39**
CATAAA	682	30.88	38.38	26.91	25.54	**13.77**
AATATA	634	24.41	36.98	22.06	17.82	**10.88**
GATAAA	528	28.11	37.31	23.26	22.15	**13.63**
AATACA	368	32.97	33.89	24.72	22.00	**15.44**
AAAAAG	342	31.18	41.76	29.41	27.76	**14.90**
ACTAAA	314	28.89	39.03	24.51	25.79	**16.58**
AAGAAA	250	31.60	36.00	26.80	26.80	**12.00**
AATAGA	100	34.00	40.00	23.00	20.00	**19.00**

Table 4 Error rates comparison between SANPolyA and DeeReCT-PolyA on mouse datasets. The best performance for each PAS motif variant was indicated in bold.

Variants	SPRET/EiJ (SP)			C57BL/6J (BL)		
	Error Rate (%)					
	Size	DeeReCT-PolyA	SAN PolyA	Size	DeeReCT-PolyA	SAN PolyA
AATAAA	17708	26.50	**15.99**	20250	25.48	**16.92**
ATTAAA	7550	25.30	**15.43**	9056	24.89	**16.52**
TTTAAA	2336	19.95	**13.61**	2688	18.19	**15.18**
TATAAA	2178	22.91	**15.56**	2518	22.44	**16.79**
AGTAAA	2224	22.88	**13.62**	2376	21.63	**13.04**
CATAAA	1432	20.53	**13.89**	1760	19.77	**16.48**
AATATA	1334	23.55	**15.29**	1528	23.23	**16.55**
AATACA	1210	21.40	**13.39**	1326	22.55	**15.76**
GATAAA	1032	17.84	**11.43**	1176	18.54	**14.54**

AAGAAA	1022	15.07	**11.64**	1126	15.81	**13.05**
AATGAA	982	18.84	**12.41**	1108	18.86	**14.44**
ACTAAA	728	19.37	**14.14**	776	20.24	**14.16**
AATAGA	494	18.64	**11.76**	536	21.24	**12.85**

We compared SANPolyA with a previous tool DeepGSR (Kalkatawi, et al., 2019). Kalkatawi et al. collected a new PAS dataset (i.e. GSR-human), and compared DeepGSR with previous tools in predicting AATAAA PAS motif in human (Kalkatawi, et al., 2019). It was reported that DeepGSR outperformed previous tools with an error rate of 13.06%. To consistent with DeepGSR, we used the model trainning and validation strategy in DeepGSR, and built SANPolyA on the AATAAA PAS data of the GSR-human PAS dataset. Results shown that SANPolyA outperforms DeepGSR in predicting AATAAA PAS in human (error rate: 6.27% vs. 13.06%).

Finally, we evaluated SANPolyA on our newly collected unbalanced SAN-human dataset. For each PAS motif variant, we calculated the average performance of SANPolyA as above. As for the unbalanced dataset, area under the precision recall curve (AUPRC) is the common metric to evaluate the model. SANPolyA generally showed decrease in performance on unbalanced dataset compared with balanced dataset (Table 5, S8),

Table 5 AUPRC performance of SANPolyA on unbalanced SAN-human dataset.

		AUPRC		
Variants	Number of True PAS	Negative : positive sample numbers 1:1	2:1	4:1
AATAAA	24509	0.904	0.832	0.691
ATTAAA	7040	0.901	0.828	0.682
TATAAA	1587	0.906	0.819	0.671
AGTAAA	1305	0.905	0.841	0.695
CATAAA	718	0.921	0.848	0.754
AATATA	678	0.881	0.856	0.673
GATAAA	550	0.909	0.849	0.736
TTTAAA	393	0.906	0.778	0.579
AATACA	353	0.878	0.824	0.73
AAAAAG	313	0.882	0.819	0.666
ACTAAA	309	0.86	0.797	0.71
AAGAAA	262	0.899	0.795	0.636
AATAGA	112	0.843	0.787	0.597
AAAACA	90	0.826	0.856	0.616
AATGAA	87	0.861	0.846	0.817
AATTAA	86	0.955	0.778	0.731
ATGAAA	78	0.953	0.84	0.746
ATAAAA	57	0.947	0.841	0.798

3.3 Leave-one-motif-out evaluation

Following a previous study (Xia, et al., 2018), we conducted a leave-one-motif-out evaluation on omni-human dataset. As mentioned in Section 2.1, omni-human dataset contains true and pseudo PAS sequences for the 12 most common PAS motif variants in human. We trained a specific SANPolyA model based on 11 PAS motif variants data, and used the remaining 1 PAS motif variant data to test the model. Obviously, the remaining 1 PAS motif variant data were not involved in the training process of SANPolyA. If SANPolyA gets good performance on the leave-one-motif-out evaluation, it implies that SANPolyA can predict the unknown PAS motif variants. For each PAS motif variant, we conducted such a leave-one-motif-out evaluation process. In the training process of each leave-one-motif-out evaluation, we chose 90% of the 11 PAS motifs data for training and the remaining 10% for validation. We compared the leave-one-motif-out evaluation results of SANPolyA with the standard results of SANPolyA, Omni-PolyA and DeeReCT-PolyA that were shown above in Table 3 (Table 6). The results of the leave-one-motif-out evaluation of SANPolyA were all worse than the standard results of SANPolyA. However, the results of the SANPolyA on leave-one-motif-out evaluation were quite comparable with standard results of previous tools. Even for some PAS motif variants, like AATAGA and AAAAAG, the results of the SANPolyA on leave-one-motif-out evaluation were better than standard results of Omni-PolyA and DeeReCT-PolyA. Our results imply that our model SANPolyA can deal with the unknown PAS motif variants that are not included in the training data.

Table 6 Error rates of SANPolyA with leave-one-motif-out evaluation on the omni-human dataset.

Variants	Size	Error Rate(%)			
		SANPolyA (leave-one-motif-out)	Omni-PolyA	DeeReCT- PolyA	SANPolyA
AATAAA	24,310	28.50	23.96	21.99	13.99
ATTAAA	7098	27.59	24.20	23.01	14.14
TATAAA	1640	28.41	25.86	23.60	15.67
AGTAAA	1306	28.56	23.07	20.21	13.39
CATAAA	682	27.27	26.91	25.54	13.77
AATATA	634	27.76	22.06	17.82	10.88
GATAAA	528	22.92	23.26	22.15	13.63
AATACA	368	27.99	24.72	22.00	15.44
AAAAAG	342	26.61	29.41	27.76	14.90
ACTAAA	314	29.30	24.51	25.79	16.58
AAGAAA	250	31.60	26.80	26.80	12.00
AATAGA	100	18.00	23.00	20.00	19.00

Table 7 The numbers of PAS variants in the dragon-human and omni-human datasets after filtering out pairs of homology sequences.

Variants	Dragon-human		Omni-human	
	Original	After filtering	Original	After filtering
AATAAA	5190	4406	24310	23048
ATTAAA	2400	1872	7098	6452
AAGAAA	1250	882	250	224
AAAAAG	1230	530	342	308
AATACA	880	446	368	306
TATAAA	780	544	1640	1366

ACTAAA	690	314	314	248
AGTAAA	670	456	1306	1164
GATAAA	460	270	528	462
AATATA	410	230	634	512
CATAAA	410	234	682	560
AATAGA	370	212	100	80

3.4 Analysis of the homology effect of the PAS datasets

We sought to examine whether homology sequences exist in collected PAS datasets. If this is the case, we tested whether homology bias PAS prediction. Inspired by a previous study (Chen, et al., 2019), we used the CD-HIT software (Fu, et al., 2012) to identify pairs of homology sequences in the PAS dragon-human and omni-human datasets with the lowest threshold 0.8. We filtered out all identified pairs of homology sequences. After filtering, the number of samples in dragon-human dataset was reduced from 14,740 to 10,396, and the number of samples in omni-human dataset was reduced from 37,572 to 34,730 (Table 7).

We retrained SANPolyA on the filtered dragon-human and omni-human datasets. We calculated the weighted average error rate of the dragon-human and omni-human datasets before and after filtering (Figure 3). The weight of every PAS variant is the sample number of the variant. Homology sequences have effects on PAS identification: the error rates become higher after filtering out pairs of homology sequences. However, the increased changes were quite small for the two datasets (~1%).

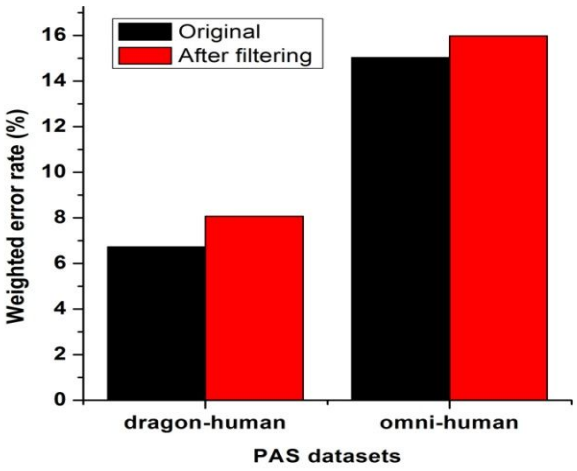


Figure 3. The effect of homology on PAS identification. Weighted average predicted error rates by SANPolyA were shown for original datasets and homology sequences-filtered out datasets.

3.5 Visualization of sequence motifs learned by SANPolyA

We visualized the convolutional filters in the first convolutional layer of SANPolyA. Each filter in the first convolutional layer can be regarded as a specific DNA sequence motif recognizer. In SANPolyA, the filters in the first convolutional layer are all 10 units long. After the input sequence is fed to the model, the filters of the first convolutional layer scan all the 10-nt long sub-sequences from the beginning to the end of the input sequence. Each filter performs convolution with the 10-nt long sub-sequence and outputs an activation value for this 10-nt sub-sequence. If a 10-nt sub-sequence has a large activation value for a specific filter, it means that this sub-sequence is more likely to be the pattern that the filter wants to find. We selected 20% of the dragon-human dataset as the visualization dataset. For each genomic sequence in the visualization dataset, we applied all 16 filters in the first layer of our model to the input sequence, and for each filter, we identified the best 10-nt sub-sequence that has the largest activation value. In this way, for all input sequences, we had many 10-nt sub-sequences for each of the 16 filters. We used these 10-nt sub-sequences to construct position weight matrices and visualized the pattern in sequence logos for each of the 16 filters. (Figure 4) by the tool WebLogo (Crooks, et al., 2004). Interestingly, many patterns we found are A-rich, indicating U-rich RNA sequence.

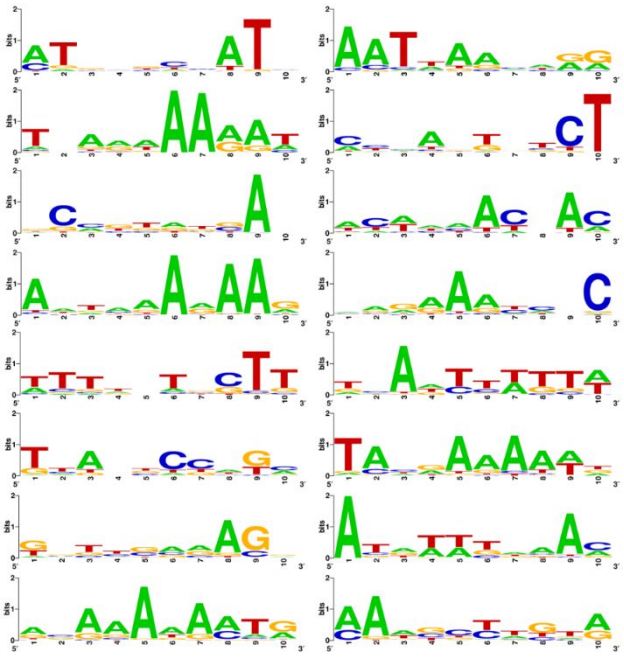


Figure 4. Visualization of sequence motifs learned by SANPolyA in sequence logos.

3.6 Predicting Poly(A) sites with no PAS motif

Analyzing the poly(A)-seq data (Wang, et al., 2017), we found that some poly(A) sites do not have PAS motifs in their upstream regions. The flanking sequence elements of PAS motif are important for the cleavage and polyadenylation. We sought to examine whether the flanking sequences around PAS motifs can help to predict the poly(A) sites with no PAS motif. We retrained SANPolyA on the omni-human dataset by only using the 100bp upstream and 100bp downstream sequences surrounding the 6bp PAS motif in every sample, that is, the 6bp PAS sequence were excluded. We used the PolyA_DB database to extract the poly(A) sites with no PAS motif in upstream regions (Wang, et al., 2017). After filtering (see details in Table S9), we got 1692 samples with no PAS motif in human genome. We extracted the 120bp upstream and 80bp downstream regions of these samples. We considered all these 1692 samples as positive (Table S10). Then we tested our retrained SANPolyA on these samples with no PAS motif. The predicted accuracy is 77.96%, indicating that SANPolyA can predict the poly(A) sites with no PAS motif.

4 Conclusions

In this study, we proposed a deep neural network-based tool, SANPolyA, to identify Poly(A) signals in human and mouse genomes. SANPolyA is

a generic model that can identify most common PAS motifs in human and mouse genomes, so that we do not need to train a specific model for every PAS motif. By comparing SANPolyA with previous state-of-art tools, we found that SANPolyA performs better than previous state-of-art methods in predicting Poly(A) signals.

Funding

This work has been supported by the the National Natural Science Foundation of China (NSFC) (Grant 61872395, U1611265), by Natural Science Foundation of Guangdong Province (2018A030313285), and also by Pearl River Nova Program of Guangzhou (201710010044).

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.

Albalawi, F.*, et al.* (2019) Hybrid model for efficient prediction of poly(A) signals in human genomic DNA, *Methods*.

Arefeen, A., Xiao, X. and Jiang, T. (2019) DeepPASTA: deep neural network based polyadenylation site analysis, *Bioinformatics*.

Beaudoing, E.*, et al.* (2000) Patterns of variant polyadenylation signal usage in human genes, *Genome Res*, **10**, 1001-1010.

Chen, F., MacDonald, C.C. and Wilusz, J. (1995) Cleavage site determinants in the mammalian polyadenylation signal, *Nucleic acids research*, **23**, 2614-2620.

Chen, W.*, et al.* (2019) i6mA-Pred: identifying DNA N6-methyladenine sites in the rice genome, *Bioinformatics*.

Chollet, F. and others (2015) Keras.

Clevert, D.-A., Unterthiner, T. and Hochreiter, S.J.C. (2015) Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), **abs/1511.07289**.

Crooks, G.E.*, et al.* (2004) WebLogo: a sequence logo generator, *Genome Res*, **14**, 1188-1190.

Fu, L.*, et al.* (2012) CD-HIT: accelerated for clustering the next-generation sequencing data, *Bioinformatics (Oxford, England)*, **28**, 3150-3152.

Harrow, J.*, et al.* (2012) GENCODE: the reference human genome annotation for The ENCODE Project, *Genome Res*, **22**, 1760-1774.

Hu, J.*, et al.* (2005) Bioinformatic identification of candidate cis-regulatory elements involved in human mRNA polyadenylation, *RNA (New York, N.Y.)*, **11**, 1485-1493.

Hunt, A.G.*, et al.* (2008) Arabidopsis mRNA polyadenylation machinery: comprehensive analysis of protein-protein interactions and gene expression profiling, *BMC genomics*, **9**, 220-220.

Kalkatawi, M.*, et al.* (2019) DeepGSR: an optimized deep-learning structure for the recognition of genomic signals and regions, *Bioinformatics (Oxford, England)*, **35**, 1125-1132.

Kalkatawi, M.*, et al.* (2012) Dragon PolyA Spotter: predictor of poly(A) motifs within human genomic DNA sequences, *Bioinformatics*, **28**, 127-129.

Lei Ba, J., Ryan Kiros, J. and E. Hinton, G. (2016) Layer Normalization.

Leung, M.K.K., Delong, A. and Frey, B.J. (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 research*, **40**, 8460-8471.

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

Magana-Mora, A., Kalkatawi, M. and Bajic, V.B. (2017) Omni-PolyA: a method and tool for accurate recognition of Poly(A) signals in human genomic DNA, *BMC Genomics*, **18**, 620.

Matis, S.*, et al.* (1996) Detection of RNA polymerase II promoters and polyadenylation sites in human DNA sequence, *Computers & Chemistry*, **20**, 135-140.

Miura, R.M., Cheng, Y. and Tian, B. (2006) Prediction of mRNA polyadenylation sites by support vector machine, *Bioinformatics*, **22**, 2320-2325.

Proudfoot, N.J. (2011) Ending the message: poly(A) signals then and now, *Genes & development*, **25**, 1770-1782.

Salamov, A.A. and Solovyev, V.V. (1997) Recognition of 3' -processing sites of human mRNA precursors, *Bioinformatics*, **13**, 23-28.

Shen, Y.*, et al.* (2008) Genome level analysis of rice mRNA 3'-end processing signals and alternative polyadenylation, *Nucleic acids research*, **36**, 3150-3161.

Tabaska, J.E. and Zhang, M.Q. (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 acids research*, **33**, 201-212.

van Helden, J., del Olmo, M. and Pérez-Ortín, J.E. (2000) Statistical analysis of yeast genomic downstream sequences reveals putative polyadenylation signals, *Nucleic acids research*, **28**, 1000-1010.

Vaswani, A.*, et al.* (2017) Attention Is All You Need. *NIPS*.

Wang, R.*, et al.* (2017) PolyA_DB 3 catalogs cleavage and polyadenylation sites identified by deep sequencing in multiple genomes, *Nucleic Acids Research*, **46**, D315-D319.

Wu, Y. and He, K. (2018) Group Normalization, *The European Conference on Computer Vision (ECCV)*, **pp. 3-19**.

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

Xie, B.*, et al.* (2013) Poly(A) motif prediction using spectral latent features from human DNA sequences, *Bioinformatics*, **29**, i316-i325.

Zarudnaya, M.I.*, et al.* (2003) Downstream elements of mammalian pre-mRNA polyadenylation signals: primary, secondary and higher-order structures, *Nucleic acids research*, **31**, 1375-1386.