**APAIQ identifies and quantifies alternative polyadenylation from RNA-seq data**

**Abstract**

Most mammalian genes use multiple polyadenylation sites (PAS) to generate RNA transcripts with varied 3’ends, which is known as alternative polyadenylation (APA). APA patterns are distinct across samples from different cell types or at different conditions, while many computational methods have been developed to characterize the sample-specific APA events using corresponding RNA-seq data. However, they are confronted with the problems of high false positives and low recall on PAS identification, whereas the method for APA quantification usually relies on the annotation. Moreover, these methods usually only focus on some specific genic regions, such as 3’ untranslated regions (UTR). Here we developed a computational tool for **APA** **i**dentification and **q**uantification (APAIQ) from RNA-seq data, which is capable of identifying the position and quantifying expression of each PAS at genome-wide. Using 3’end-seq data as the benchmark, we show that APAIQ outperforms the preexisting methods on both PAS identification and quantification, such as DAPARS2, xx, xx and QAPA. More importantly, result by using APAIQ on RNA-seq data is a faithful recurrent of that directly from 3’end-seq data. Finally, applying our method on 424 RNA-seq samples from an HCC cohort, we identified xx highly reliable PAS, including xxx novel and tumor-associated events, indicating the capacity of APAIQ to facilitate APA studies with large-scale RNA-seq dataset. APAIQ is available as a package at BIOCONDA.

**Introduction**

In eukaryotes, the transcription termination is mediated by cleavage of the nascent RNA and followed with the synthesis of poly-adenosines (poly-A) to the 3’end of the RNA, which is known as polyadenylation. This process is determined by a series of RNA binding proteins (RBP) that recognizes cis elements surrounding polyadenylation site (PAS). PAS motif, hexamers located with 15~40nt upstream of the cleavage site, is one of the most important core elements for PAS definition (Ref). They include AAUAAA that present in more than half of the PAS, and its variants (AUUAAA, xxx et al), which has been found in nearly 40% of the PAS in human and mouse genome (Ref). UGUA at the downstream of PAS is another core element for PAS definition, while other auxiliary elements located within 100bp flanking PAS also contribute to the formation polyadenylation (Ref).

Taking advantage of these elements, numerous computational methods have been developed for PAS prediction. The early studies focused on features obtained from statistical analysis of sequences surrounding the PAS. For instance, Hu et al identified 2 core elements and 4 auxiliary elements at 100bp upstream region, and 4 core elements and 5 auxiliary elements at 100bp downstream region of PAS by analyzing large number of PAS on human and mouse (Hu et al). Using these features, support vector machine (SVM) was applied for PAS prediction (Cheng et al. 2006. Bioinf). Later, a hidden Markov models (HMM) was introduced to extract latent features and fed them to an SVM model (Xie et al). More recently, with the development of deep-learning models, convolutional neural network (CNN) has been implemented to extract sequence features automatically, which greatly improved accuracy of the PAS prediction (Xia et al).

However, most mammalian genes use multiple sites for polyadenylation to generate RNA isoforms with varied 3’ends (ref). This alternative polyadenylation (APA) process not only enables single gene to encode multiple protein isoforms, but also greatly increase the complexity of gene regulations via different 3’ untranslated regions (3’ UTRs). In general, the usage choice of different PAS for each gene is variable across different types of cells. For instance, neuron cells tend to use the distal PAS to generate long isoform, while proliferated cell express short isoforms by using proximal PAS (Ref). Moreover, even the same type of cells, APA alterations has also been observed at different conditions (Ref). These suggest APA is tightly regulated across different cell types and biological conditions. Dysregulation of APA is associated with human disease, including cancer. It has been shown that APA mediated 3’UTR shortening activates oncogenes expression via escaping from microRNA regulation, while it could also inhibit tumor suppressor in trans by disrupting the competition of microRNA binding (Ref). These suggesting it is very important to identify and quantify the used PAS for each gene across different cells and samples.

However, different cells and samples are usually share the identical DNA sequence, while the usage of the PAS for each gene could be distinct. The computational methods solely using DNA sequence is very unlikely to predict and quantify this sample-specific used PAS. On the other hand, many experimental technical have been developed for genome-wide identification and quantification of sample-specific APA events by enriching the 3’end of RNA transcripts, followed with high-throughput sequencing (3’end-seq). However, these methods are rather laborious and more time and material-consuming comparing to convention RNA sequencing (RNA-seq). More importantly, unlike conventional RNA-seq, 3’end-seq data is almost impossible to be used for studying other posttranscriptional RNA metabolic process beyond APA, such as splicing and RNA editing.

Accordingly, many computational methods have also been developed for identification and quantification of APA from RNA-seq data. Almost all of them are designed to detect the drops in RNA-seq coverage along the genome. However, due to the high fluctuation of the coverage potentially caused by heterogenous mappability, as well as amplification-efficiency of different RNA fragment in sequencing library preparation, these methods frequently face the problems of high false positives and low recall on PAS identification. Just recently, a study benchmarking multiple computational tools for APA analysis, found that none of them could achieve >50% of recall and FDR is usually as high as 30 to 50% (Ref). Moreover, the previous computational methods mainly focus on APA within 3’UTR, whereas it’s very challenge to be applied in other genic regions, which might be due to that dramatically drops of RNA-seq coverage across exon-intron boundary could impact the prediction. This limits their application for APA identification beyond 3’UTR, such as intronic polyadenylation (IPA) that has been revealed widespread in leukemia to inactivate tumor suppressor by generating truncated protein isoforms (Ref). In addition to identification, methods for APA quantification usually rely on the annotation of PAS, but the current PAS annotation are mainly derived from common cell types or biological samples. There might be tremendous of unannotated PAS in rare cell types or very heterogenous biological samples, such as primary tumor samples.

Here, we developed APAIQ, a computational method which is capable of sample-specific **APA** **i**dentification and **q**uantification from RNA-seq data. By integrating DNA sequence and RNA-seq coverage into a deep-learning model, APAIQ can accurately and comprehensively identify PAS genome wide in each sample. It outperforms the existed method, either purely based on DNA sequence, or RNA-seq coverages, with much higher recall and lower false positives. Meanwhile, beyond the 3’UTR, APAIQ is able to identify APS globally. In addition, APAIQ is also capable of APA quantification in terms of faithful predict both the expression level and relative usage of each PAS in genes used multiple PAS. Finally, we applied APAIQ on large-scale RNA-seq data from hepatocellular carcinoma (HCC) patients and identified hundreds of tumor specific APA events, including IPA events, which have not been reported before. Overall, APAIQ is powerful tool for accurate genome-wide APA analysis using RNA-seq data, which will enable to build more comprehensive APA landscape with the existing RNA-seq data across different cells, tissues and pathological status, including cancer.

**Results**

**A hybrid deep-learning model for PAS identification**

To identify expressed PAS in each sample with RNA-seq data, most previous computational methods were designed to capture the transitions in RNA-seq converge. On the other hand, computational methods solely using DNA sequence could only predict PAS in general, regardless of the expression specificity across different samples. Here, we develop a hybrid deep-learning model for samples-specific PAS identification using both converge from RNA-seq data and the corresponding DNA sequence. We performed RNA 3’end sequencing on four cell lines, including K562, HepG2, THLE2 and Hep3B, to comprehensively characterize the expressed PAS in sample-specific manner. Eventually, we identified an average of xx PAS in each cell line with decent expression level (Methods, Table 1). Using them as positive dataset, and random loci apart from the PAS as negative dataset, we trained a model for binary classification in each cell line based on CNN (Fig. 1a). We found that the RNA-seq coverage indeed tended to be dropped at the downstream of PAS comparing to upstream (Fig .1b). In addition, xx% of positive dataset have canonical PAS motif (AAUAAA) and its variants, while only xx% of negative dataset have the motifs (Fig. 1c). These further suggested that integrating DNA sequence with RNA-seq converge might have a synergistic effect in sample-specific PAS identification.

**APAIQ predicts PAS accurately in transcriptome wide**

To evaluate the feasibility of integrating two features for PAS identification, we applied a cross validation approaches, in which we trained the model using 1/5 of the datasets and made predictions on the rest. Just as expected, the integrated model achieved xx TPR/recall with FDR less than xx, while models only using DNA sequence or RNA-seq coverage have much worse performance (TPR = xx, FDR = xx, Table 1).

Even though the binary classification model could achieve good performance in the determination of true or false PAS, applying it on genome-wide PAS identification in practical remains challenge since the positive and negative datasets are highly unbalanced. For instance, among 3 billion loci in human genome, only xx of them are truly expressed PAS (positive), while the rest 9x% are negative. To this aim, we extended the model to genome scanning. In brief, we did prediction at each locus and any locus within xx bp away from true PAS were considered as true positive. (Please write a summary about how you did the scanning). Furthermore, even the model could achieve 99% of precision in the binary prediction, the final genome scanning could still get millions of false positives, resulting in extremely high FDRs. To solve this problem, we first limited the loci for the prediction by requiring it covered by RNA-seq reads. In this way, we reduced the total scanning loci from 3 billion to xx million. Additionally, we introduced a data argumentation strategy to increase the complexity of the training dataset, as well as a circular training strategy during scanning, in which we repeatedly replace the false positive predictions, which further reduced the false positives.

Eventually, our model was able to identify ~70% of the used PAS (recall of TPR: true positive rate), while only less than 10% of the un-used PAS was falsely predicted (FPR: false positive rate). Interestingly, we found that around 75% of PAS in SNU398 and THLE2 could be detected, while only 60% of PAS in HepG2 and K562 were detected with FDR < 10% (Supp Figs. 1a and 1b). This could be due to RNA-seq data depth? polyA-seq depth?. Even though to obtain ~80% recall, FDRs could be as high as 30% in our prediction, many of them might not be real false positives as they are expressed and annotated, whereas only the expression levels did not pass our threshold (Fig. 2a). To further inspect this, we further grouped annotated PAS with a series of expression levels. Indeed, our model exhibit much stronger capacity of the identification of PAS with higher expression, while it’s not so powerful for lowly expressed PAS (Fig. 2b). We further checked the position of our identified PAS and surprisingly they are very closed to the true PAS, in which >80% of the identified PAS are within 25bp away from the annotation (Fig. 2c). As shown in Fig. 2d as an example, our model was successfully identified PAS a from gene X, which was expressed in xx, xx, but not in xx, while PAS b was expressed in xx xx, but not in xx. Taken all these results together, they suggested that our method is capable of accurate PAS identification using RNA-seq data in genome wide scale.

**APAIQ outperforms other methods on PAS identification**

Since our model showed good performance on PAS identification, we next benchmarked it with several previous published methods, including DaPars2, mountain climber, xx, and APxx. Among them, xx and xx were based on the transition pattern in RNA-seq coverage, while xx and xx relied on DNA sequence, while APXX combined RNA-seq and DNA sequence, but only capable of binary classification. As shown in Fig. 3a, with the same FDR, our method identified the largest number of truly expressed PAS. Moreover, the distance between the identified PAS and true PAS, is smallest in our method comparing to the other 4/5? (Fig. 3b).

Moreover, as xx and xx were designed to detect APA events within the 3’UTR, we classified the expressed PAS into xx groups based on their genomic localizations, including terminal exon (including 3’UTR), upstream exonic regions, upstream intronic regions, and intergenic regions. We found that our method outperformed all these published methods in each genomic region (Table S2). As shown in Fig. 3c, our method identified a PAS within intronic regions, while all other method failed to do so.

To dissect the contribution of RNA-seq converge and DNA sequence to the model, we also compared the integrated model with coverage- and DNA-only model. We found that, using two features not only improved precision, but also reduced FDR in the prediction. An example was shown in Fig. 3d, using DNA-only model, identified an annotated PAS that are not expressed, while adding the RNA-seq coverage features, the integrated model was successfully discarded this PAS. Another example in Fig. 3e, RNA-seq coverage identified a false PAS from gene x due to the drastic drop of coverage, while there is none of sequence features supporting this PAS, which was also successfully predicted as false by the integrated model (Fig. 3f). In addition, some PAS were neither identified in DNA-only, nor coverage-only model, but finally identified in the integrated model, further confirmed a synergistic effect by combining these two features for PAS identification.

**Transfer abilities of APAIQ across different cell lines**

To examine whether the deep-learning model could be applied on different samples, we tested the model across different cell lines. Simply, we trained the model in each cell line and made predictions on the samples from other three cell lines. We found that it showed relatively good transferring abilities. For instance, the model trained in xx, could still get xx recall with xx precision in xx cell lines (Fig. 3e/Fig. S). To further tested whether the model could be even applied on another species, we tested the model on another dataset from mouse (Fig. 3f/Fig. S). Still, the model showed good performance, indicating a good generalization ability of our method on different samples.

**Accurately quantification of APA by APAIQ**

After the identification of PAS based on RNA-seq data, we further asked whether is possible to quantify the expression of each PAS based on RNA-seq data. To do so, we introduced a regression model, in which we considered the expression of the PAS (Y) is a function of the coverages at each locus within 100bp flanking the PAS (X). The expression level could be estimated by 3’end-sequencing data and the coverage at each locus were derived from RNA-seq data. Our prediction achieved an average Pearson correlation 0.77 in four cell lines (Fig. 4a).

Among xx genes with expressed PAS based on 3’end-seq data, xx genes have more than one PAS. We further examined the ability of our method for APA quantification. For each PAS, we calculated their relative usage using its expression divided by the total expression of this gene. The usage of PAS showed a typic bimodal distribution, in which most of PAS were either predominant PAS or minor PAS within a gene (Fig. 4b). Our method could still predict the relative usage of these PAS faithfully with a Pearson correlation as high as 0.7x, which is much accurate than previous published method QAPA, xx, and xx (Fig. 4c).

**Applying APAIQ on a large-scale RNA-seq data identifies tumor-associated APA events**

Finally, to test our method, we applied it on xx RNA-seq samples from The Cancer Genome Atlas Liver Hepatocellular Carcinoma (TCGA-LIHC) (Fig. 5a). In total, xx PAS were identified and xx of them are overlapped with annotation. Among them, xx are within terminal exon/3’UTR and xx of them are located at upstream exon and intronic regions (Fig. 5b). To further evaluate the quality of the identified PAS, we first checked the frequency of canonical PAS motif (AAUAAA) and its variants and found that xx% of the identified PAS has the motif. In addition, the nucleotide composition around the identified PAS share a similar patter as the results from 3’end sequencing data reported in previous study (Fig. 5c).

Next, we also quantified the expression of each PAS and comparing them between samples from tumor and normal. Finally, we identified xx PAS which are significantly upregulated in tumor, including xx were only detected in tumor samples. We also correlated the expression of each PAS with the patient overall survival rate, and xx of them are associated with poor prognosis, while xx of them are associated with good survival. These xx PAS are from nn genes and very interestingly, expression of many of these genes are not associated with prognosis, while the PAS was, suggesting different isoforms with the varied 3’end might have distinct functions in cancer progression. As shown in Fig. 5e, an intronic PAS from gene X were identified in xx% of tumor samples but absent in all normal samples. This IPA events might produce a truncated protein isoform, which might contribute to liver cancer development.

**Discussion**

**Methods and Materials**

**Cell lines and high-throughput RNA sequencing**

**Sequencing data processing**

RNA-Seq data from four cell lines were aligned to human reference genome (hg38) with the transcriptome annotation (https://www.gencodegenes.org) using STAR (version 2.7.0). We applied Encode standard parameters (link) for the alignment and only the unique mapped reads were kept for the further analysis. The parameter ‘- -outWigType bedGraph’ from STAR was used to generate files of the RNA-seq coverage in bedGraph format, in which the coverage at each genomic locus were normalized to reads per million (RPM) using the total uniquely mapped reads.

For 3’end sequencing data, using the same criteria as previous study[ref], Illumina sequencing adaptor at the 3’end of the read and another 12nt sequence from random primer during the second-strand cDNA synthesis were removed. The first nucleotide at the 5’end of each read was further trimmed and the derived clean reads with length longer than 15nt were aligned to reference genome (hg38) using STAR (version 2.7.0).

**Positive and negative dataset preparation**

To get datasets for training and testing the deep-learning model, we first build the general annotation using xx PAS from polyADB3 and xxx PAS extracted from the transcript end based on Gencode annotation. Next, in each cell line, we mapped the reads from 3’end sequencing data to the annotation with ‘bedtools?’ to get the expressed PAS in each sample (the sample-specific PAS). In brief, for each PAS, any reads with 3’end located within 25 bp were counted and then normalized to RPM using total unique mapped reads. If two annotated PAS were located within 50bp, only the one with higher expression was kept. Next, in each sample, by overlapped the PAS in with gene annotation, we calculated the relative usage of each PAS by divided the expression of the PAS to the sum of all the PAS within the same gene. Any PAS with expression level lower than 0.05 RPM and usage lower than 0.05 was discarded.

For each retained PAS in each cell line, we extracted DNA sequence and RNA-seq coverage from 100bp upstream to 100bp downstream regions. To avoid discrepancy between RNA-seq and 3’end-seq data from the same sample/cell line, we further filtering out the PAS by requiring the average RNA-seq coverage at 100bp upstream should be no less than 0.05 RPM. In this way, we finally got ~20,000 true samples-specific PAS as positive dataset in each cell line (Supplementary Table x). The same number of sites with average RNA-seq coverage no less than 0.05 RPM at 100bp upstream regions, meanwhile it’s at least 50bp far away from any true PAS, were randomly selected from the genome as negative dataset.

**A hybrid deep-learning model using both DNA sequence and RNA-seq coverage**

As shown in Figure 1a, we built a hybrid deep-learning model that contains two independent convolution neuron networks, which take DNA sequence and RNA-seq coverage as input, respectively. The DNA sequence underwent one-hot encoding to 201x4 matrix and the normalized RNA-Seq coverage (RPM values) was convert to 201x1 matrix. Both of them went through a convolution layer consisting of 32 filters with kernel size 6. These followed by a group normalization layers with group size 4. A rectified linear unit (RELU) was applied to the normalized results as the activation function. After a max-pooling layer with pooing window setting as 6, features are flattened into one dimensional array and fed to the fully connected layer. Then, two features were concatenated together and fed into another fully connected layer followed by a softmax activation function to approximate the probability function. The final output is a prediction score between 0 to 1, while dropout was introduced after max-pooling layer as regularization to reduce over-fitting problems.

**Training and evaluation of the deep-learning model on binary classification**

In each sample/cell line, we trained our deep-learning model using xx% of the positive dataset and negative dataset and then evaluate the model in the rest of the dataset. We measured a series of metrics, including accuracy, recall, false positive rate (FPR), false positive discovery rate (FDR) and F1 score, for the evaluation. Using the same dataset and approaches, we compared the performance and other published methods using these metrics.

**An improved model for genome scanning**

Since our model achieved excellent performance on binary classification (Table S1), we would like to apply it for genome scanning. Unlike the binary classification, in which the positive and negative dataset are balanced, the whole genome contains much more negative datasets comparing to positive datasets. Specifically, among the 3 billion loci in human genome, only ~20,000 of them are true PAS in each sample. Even though we restrict the scanning regions where it is covered by RNA-seq reads, there are still xx million sites in average in each sample. Moreover, many of these negative sites might be falsely predicted as our model haven’t seen/learned any features from them due to only ~20,000 of them were randomly selected to train the model.

To solve this problem, we introduced a data augmentation strategy, in which we set the training epoch as 500 and the mini-batch size as 32. In each epoch, the site randomly shifted from the true PAS from -12 to 12bp was considered as positive dataset, while the site xx was considered as negative. To evaluate the model in each epoch, we split the genome into blocks with length about one million bp and we got an average of xx blocks in each sample. These blocks were further divided into five groups and a five-folds cross validation was applied for the evaluation.

Eventually, we selected the model with highest accuracy, and utilized it to scan the genome with a stepwise one base pair. Any site with coverage lower than 0.05 RPM at the upstream 100bp was ignored. In this way, for each site, we used the window from upstream 100bp to downstream 100bp as input and obtained a score between 0 and 1. As the model was trained by randomly shifting from -12 to 12bp to the true PAS, sites closed to the true PAS would also get a relative high score (>0.5). To further find the precise position of the PAS and reduced the total number of the predicted sites, we introduced a clustering method to converting site-based score (Sc) to cluster-based score (Cc). In brief, we firstly scanned the prediction score at each site in the forward direction based on genomic coordinates. The initial Cc was set as 0 and any continues sites with prediction score higher than 0.5 were merged into the cluster and Cc would be accumulated with the prediction score of the site, while any sites with score lower than 0.5 would give a penalty P to Cc. The cluster was ended when Cc dropped to 0 and the site obtained maximum CC within each cluster was reported as the peak summit. We found that these summits showing a systematic bias toward to the downstream of the true PAS. To correct this bias, we repeated the scanning in the reverse direction based on genomic coordinates, which reported summits bias toward to the upstream of the true PAS. We finally used the middle position between the forward scanned summit and the reverse scanned summit as the putative PAS.

a The transcriptome was split into small blocks with length about 1e6 bp. Specifically, we randomly partition the blocks into five folds and used four of them for training and one for validation. We set. Data augmentation was applied during the training, each PAS random shift from -12 to 12 bp of the ground truth position during each epoch. We saved the model that has the highest accuracy on validation set and used this model to scan the whole transcriptome of window length as 201 with sliding step 1. Item with upstream coverage less than 0.05 RPM were filtered and ignored. Each item had a prediction score between 0 and 1. We considered item with prediction score higher than 0.5 as PAS containing and PAS not in the middle of the window but not too far also had prediction score higher than 0.5. Therefore, we collapse continuous position in transcriptome as higher than 0.5 one PAS and predict the location of PAS. This algorithm goes through the genome in two direction (from forward strand to reverse strand and from reverse strand to forward) to correlate the system bias. If a position was predicted as true, the prediction score was added to sum. If false, one point penalty is applied. We find the maximum sum in two direction and if they both are greater than a threshold T, the position of the PAS will be average the two maximum sum position. Even though we collapsed the transcription, we still got too many false positive by directing applying the training model. Therefore, we randomly replaced half of the negative dataset with the false positive PAS to increase the complexity of the training set and trained again. After, four iteration training, the recall the precision converged. We kept the specific model with best F1 score.

**Data availability**

Raw 3’end-seq data of four human cell lines (K562, HepG2, SNU398, THLE2) are deposited in NCBI Sequence Read Archive (SRA) (<http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi>) under accession number SRPXXXXXX. The public RNA-seq from THLE2 and SNU398 cell lines are also available at SRA under accession number PRJNA495931 and PRJNA562266, respectively. RNA-seq data of HepG2 is available at ENCODE (www) and the samples with library ID ENCLB471LNG and ENCLB352YLJ were used in this study. In-house RNA-seq data of K562 cell line is also available in SRA under accession number XXX. The public RNA-seq and 3’end-seq data from mouse fibroblast are available on ENA under accession number PRJEB7211. The annotation of PAS are downloaded from PolyA\_DB3 (<https://exon.apps.wistar.org/PolyA_DB/v3/>) and Gencode (<https://www.gencodegenes.org/>) .

**Code Availability**

The open source code of APAIQ is freely available at <https://githun.com/ijayden-lung/APAIQ> and the compiled package can be found on bio-conda.