Termin(A)$_n$tor: Polyadenylation Site Prediction Using Deep Learning Models

Chen Yang[1,2], Chenkai Li[1,2], Ka Ming Nip[1,2], Rene L Warren[1], Inanc Birol[1,2*]

1. Canada's Michael Smith Genome Sciences Centre, BC Cancer, Vancouver, BC, Canada

2. University of British Columbia, Vancouver, BC, Canada

* Corresponding author

ABSTRACT

As a widespread RNA processing machinery, alternative polyadenylation plays a crucial role in gene regulation. To help decipher its underlying mechanism and understand its impact, it is desirable to comprehensively profile 3'-untranslated region cleavage and associated polyadenylation sites. State-of-the-art polyadenylation site detection tools are influenced either by library preparation or manually selected features. Here we present Termin(A)$_n$tor, a deep neural network-based profiling pipeline to predict polyadenylation sites from RNA-seq data. We show how Termin(A)$_n$tor outperforms competing tools in sensitivity and precision on experimental transcriptome sequence data. We also demonstrate applications of Termin(A)$_n$tor with both short-read and long-read sequencing technologies.

KEYWORDS

Deep learning, Neural network, Alternative polyadenylation, RNA-seq

BACKGROUND

Since poly(A) sequences were first discovered on the 3' ends of eukaryotic mRNAs 40 years ago (1), numerous studies have contributed to the understanding of the mechanisms, evolution, regulation, and impacts of polyadenylation (poly(A)) (2–5). As an essential step in mRNA maturation, RNA polyadenylation involves two phases: 3' end cleavage of nascent RNA and addition of a poly(A) tail. Because most nascent RNA in eukaryotes have more than one possible 3' end cleavage site (CS), mRNAs with different 3' ends are formed following these two coupled processes (6). This phenomenon is also known as alternative polyadenylation (APA), which results in transcripts with different 3' untranslated regions (UTR). In recent years, APA modulation under different physiological and pathological conditions , and recent advances have shed light on its implications in some diseases, especially cancer (5,7–9).

The importance of profiling poly(A) sites

APA and together alternative splicing, both have considerable impact on the modulation of gene expression and contribute to the transcriptome complexity. To date, the catalogue of profiled poly(A) sites, especially cancer-specific ones, is far from complete. Ensembl annotation version GRCh38.94 recorded 63,620 poly(A) sites for 19,907 protein coding genes in human, and among them, 73.13% genes can produce two or more APA isoforms (Supplementary Figure S1) (10). The most recent poly(A) database, PolyA_DB3, compiled 24 human samples and cataloged 108,042 poly(A) sites for 20,998 genes (11). The substantial improvement in 3' end annotation compared to Ensembl suggests the sample-specific usage of poly(A) sites and highlights the incompleteness of the current annotation. Moreover, usage of these poly(A) sites is dynamically regulated under different developmental stages and pathological changes, as elucidated in previous studies (5,7,12,13). Profiling poly(A) sites with respect to biological conditions serves as the first step towards deciphering the underlying mechanism of APA-mediated gene regulation. Hence, it is desirable to incorporate a fast and robust poly(A) site characterization tool into standard transcriptome analysis pipelines.

Current methods and limitations

In the past decades, continuous efforts have been made to annotate 3' ends in the genome and predict poly(A) sites in the transcriptome, using methods designed for two major sequencing protocols: 3' end sequencing and poly(A)-selected RNA-seq. 3' end sequencing methods are specialized high throughput sequencing techniques that are developed to overcome the low sequence complexity nature of the 3' end of transcripts, including TAIL-Seq, PolyA-Seq, PAS-seq, PAT-seq, etc. (14–17). Essentially, only the 3' ends of transcripts are interrogated by primers and sequenced, in combination with subsequent sequence analysis pipeline. Comparing to RNA-seq, a critical advantage of these protocols is their high sensitivity in detecting poly(A) sites from lowly expressed transcripts. With the help of 3' end sequencing technologies, several poly(A) site databases have been established and the current 3' end annotation is significantly improved as a result. Though powerful, they require specialized library preparation, which can be costly and laborious and have not yet been widely exploited in genetic researches because their application is restricted to poly(A) detection.

In contrast, RNA-seq is the established sequencing method of choice for transcriptomics. It is not only ideal for poly(A) site profiling on a per-sample basis, but also for systematic APA analysis and retrospective studies of large cohort RNA-seq data. Poly(A) site prediction and annotation tools designed for RNA-seq have demonstrated that with decent coverage (Read coverage >= 2). RNA-seq reads contains sufficient information for the identification of poly(A) tails and discovery of novel poly(A) sites (18,19). Generally speaking, these tools can be classified into three approaches: read-evidence based, expression level based, and machine learning based.

Read-evidence based approaches, represented by KLEAT and ContextMap2, search for nontemplated adenosines (A's) in aligned reads or *de novo* assembled transcripts as evidence to determine the location of poly(A) sites (18,19). Although these tools are able to discover novel poly(A) sites with high resolution, they may lose sensitivity in low coverage sequencing libraries because of the inadequate read evidence support (i.e. insufficient reads mapping to one site).

For approaches based on expression levels, poly(A) site prediction is actually a by-product of APA analysis. Tools adopting this approach such as DaPars, APAtrap, and QAPA, all rely on differential expression in the 3' UTR of different samples (20–22). Due to the low sequence complexity of 3' UTRs, read coverage in these regions may not reflect true expression levels, which will impact differential expression analysis and associated statistical tests. As a consequence, these methods may fail to detect novel but lowly expressed poly(A) sites. In addition, only poly(A) sites involved in significant APA events are reported, thus these tools are not capable of profiling all poly(A) sites in given samples.

Another angle to address the prediction problem is through machine learning (ML) approaches, and several such models have been proposed. These include more traditional ML methods like support vector machine and Hidden Markov Model, but also the latest deep learning models, such as DeeReCT-PolyA and DeepGSR (23–26). Deep learning models generally outperform

traditional classifiers because they abandon manually selected features. Raw sequences are directly fed in and hidden features may be learnt and modeled. The latter two tools mentioned above have been reported to show around 90% accuracy on test sequences (23,24). However, these models are only trained to distinguish whether a poly(A) signal (PAS, a conserved hexamer motif) is real, but in reality, a small proportion of functional human poly(A) site does not require PAS (27,28). The performance of these tools on experimental RNA-seq data has not yet been explored. At present, these approaches overlook poly(A) sites with noncanonical *cis*-acting elements, which would likely negatively impact their performances on experimental data.

Termin(A)$_n$tor

Despite the availability of various tools, comprehensively and efficiently profiling all poly(A) sites in RNA-seq samples still remains challenging. In this study, we present Termin(A)$_n$tor, a deep neural network (NN) model for fast and accurate poly(A) site recognition that is independent of the existence of a PAS. Termin(A)$_n$tor takes fixed length sequence as input and performs three-label classification to determine whether the sequence contains a poly(A) site, a non-polyadenylated CS, or no sites. In addition, we also propose a profiling pipeline that generates precise prediction of poly(A) sites using raw RNA-seq data as input. Its performance was cross-validated on two data sets of sequences (human and mouse) to select the optimal input sequence length and model hyperparameters. We have benchmarked our pipeline against competing tools on experimental RNA-seq libraries generated from the Illumina paired-end and PacBio Single Molecule, Real-Time (SMRT) Sequencing platforms.

RESULTS

Performance of Termin(A)$_n$tor using different sequence lengths

Here we present Termin(A)$_n$tor, a deep based classifier that is trained on experimentally validated DNA sequences containing three types of sites: poly(A) site, non-polyadenylated CS, or no sites. Termin(A)$_n$tor computes the probability of a test sequence belonging to one of the three classes. We generated two data sets of sequences, human and mouse, based on poly(A) site databases and Ensembl annotation, and comprehensively evaluated the performance of Termin(A)$_n$tor (Methods).

With the cross-validated optimal hyperparameters, we finalized the architecture of Termin(A)$_n$tor as shown in Methods. Input sequences from all three classes have equal lengths, each of which has three parts: upstream sequence, poly(A) site, and downstream sequence. Upstream sequences are from corresponding transcripts or genomic regions if no transcript is found in that region, while downstream sequences are genomic. Since there is no general consensus from previous work on how the sequence length influences prediction accuracy, we built models with different lengths to evaluate its effect on performance. Eight upstream lengths (200, 160, 120, 100, 80, 60, 40 and 20 nt) and five downstream lengths (200, 100, 40, 20 and 0 nt) were chosen to create a total of 40 models, which were trained to explore the impact of sequence length on poly(A) site prediction (Figure 1). In general, all models show clear separation between Poly(A) sites and the other two labels (Supplementary Figure S2). The

accuracy of all 40 models is above 87%, including the one trained with only 20 nt long upstream sequence. As expected, models trained with sequences containing both upstream and downstream sequences perform better than the ones trained with upstream sequences alone. However, given the same upstream sequences, models trained with downstream sequences longer than 20 nt perform comparably to each other. The 28 models trained with upstream sequences in {40, 60, 80, 100, 120, 160, 200} and downstream sequences in {20, 40, 100, 200} have an average accuracy of 94.50% ± 0.0015. More interestingly, longer downstream sequences do not necessarily lead to better performance. Models trained with 200 nt downstream sequences have slightly lower accuracy than the ones trained with shorter lengths, suggesting that some polyadenylation recognition patterns maybe obscured by sequence motifs further downstream.

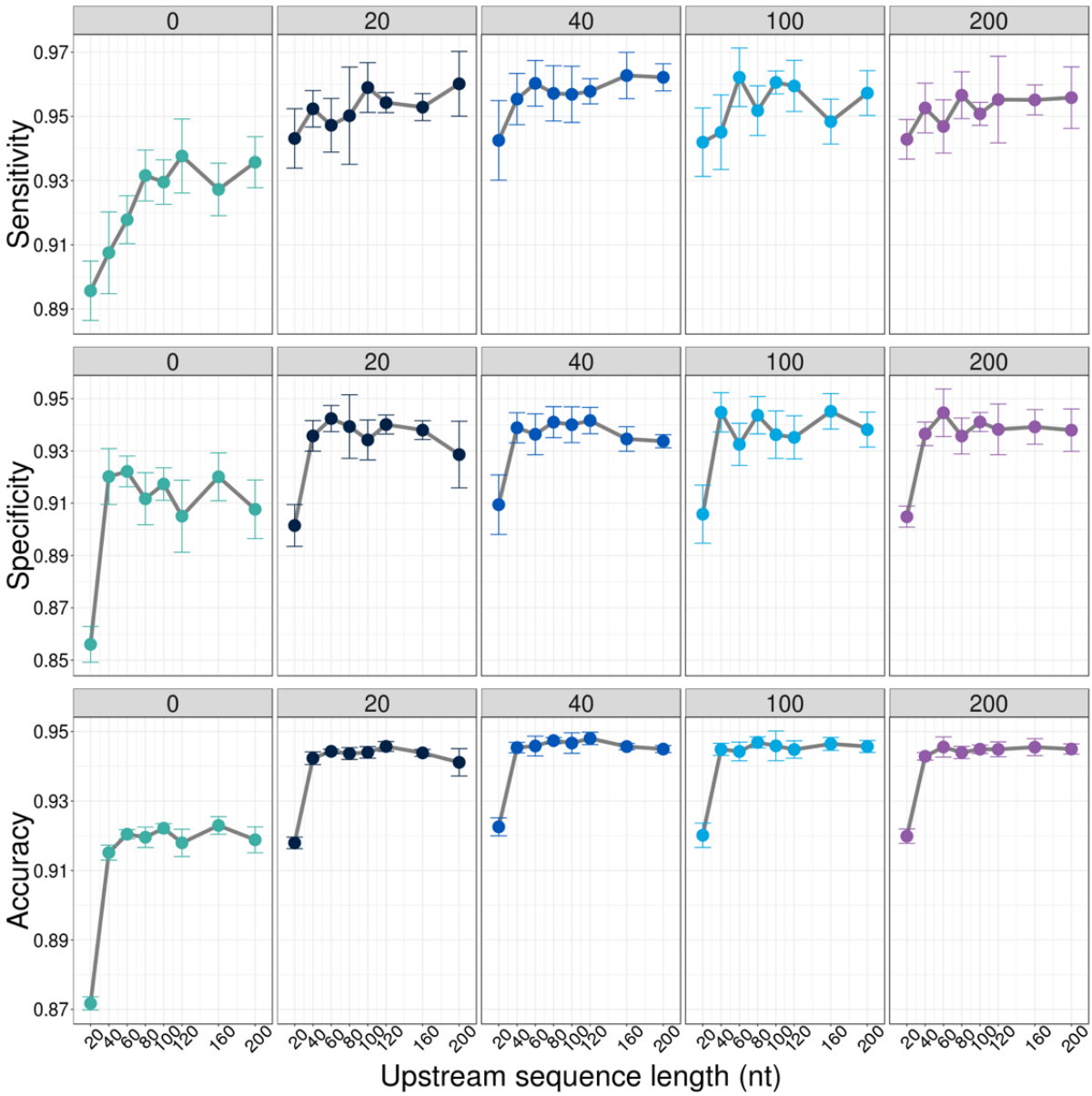

Figure 1. **Model performance on human data set**. Each facet plot represents one downstream

sequence length, and the x-axis of each facet plot shows the upstream sequence length. Each data point is the mean of 5-fold cross-validation and the error bar is standard deviation.

On the other hand, given the same downstream sequence, the performance of models plateaus when the upstream sequence is 40 nt or longer. Similarly, longer upstream sequences do not guarantee better performance either, especially when the upstream length is much longer than the downstream length. For example, the model with 120 nt upstream sequence and 20 nt downstream sequence is nearly 0.5% more accurate than the one with 200 nt upstream sequence and 20 nt downstream sequence. This decrease is mainly due to the drop in specificity, which means the model has more false positive predictions with longer upstream sequence.

Tool comparison on experimental data

In order to demonstrate the application of Termin$(A)_n$tor on poly(A) site profiling, we designed a prediction pipeline and applied it on two human RNA-seq samples, Human Brain Reference RNA (HBR) and Universal Human Reference RNA (UHR). As done previously, we trained Termin$(A)_n$tor with 40 length combinations, applied all 40 models on UHR sample, and computed the Pareto frontier based on sensitivity and precision. The sensitivity of Termin$(A)_n$tor is positively correlated with upstream sequence length, while there is no clear trend between the downstream sequence length and performance metrics (Supplementary Figure S3). For demonstration purpose, we chose ± 100 nt model for performance benchmarking and comparison.

Next, we generated two models in Termin$(A)_n$tor by training on human and mouse data sets separately with ± 100 nt sequences. We applied these two models on the UHR and HBR samples and compared the performance with KLEAT, DeeReCT-PolyA and DeepGSR (Figure 2A, Supplementary Figure S4A). Poly(A) sites predicted by these methods were compared against Ensembl annotation to calculate the sensitivity and precision. It is important to note that DeeReCT-PolyA and DeepGSR are binary classifiers while Termin$(A)_n$tor outputs a probability value that can be tuned in favour of either sensitivity or precision. Nevertheless, Termin$(A)_n$tor trained on human sequences consistently outperforms all competing tools in both metrics, except for one data point, which corresponds to the raw, filter-less KLEAT predictions. However, their precision is too low (35.90% and 25.52% for UHR and HBR samples, respectively) to be used for high-confidence predictions. Although KLEAT performs comparabaly to Termin$(A)_n$tor on UHR sample, its performance is surprisingly low on HBR sample regardless of what read evidences are used to filter the its raw predictions. It is also worth mentioning that the four models built by DeeReCT-PolyA and DeepGSR all have a precision lower than 50%, while precisions of the Termin$(A)_n$tor human model (at probability = 0.5 cut-off) are 55.59% and 57.72% for UHR and HBR samples, respectively.

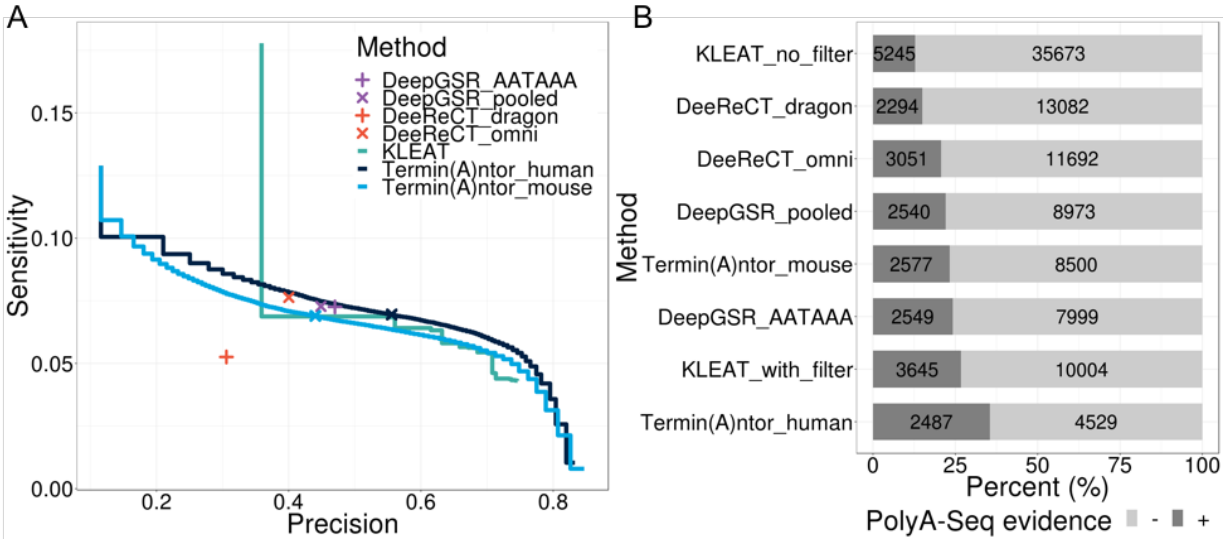

Figure 2. **Performance comparison on HBR sample.** (A) Sensitivity and specificity of poly(A) sites predictions from Termin(A)$_n$tor, KLEAT, DeeReCT-PolyA and DeepGSR on Ensembl annotated poly(A) sites. The two pre-trained models of DeepGSR are the one with sequences containing only the hexamer AATAAA, and the one with sequences containing all hexamers pooled together. Two pre-trained models of DeeReCT-PolyA are the one with dragon data set and the one with omni data set described in the DeeReCT-PolyA publication. Two pre-trained Termin(A)$_n$tor models are the one with human data set and the one with mouse data set. The navy /blue crosses on Termin(A)ntor human/mouse model represent probability = 0.5 cut-off, respectively. (B) Poly(A) sites that are missing from Ensembl annotation were compared to the ones predicted by PolyA-Seq.

Since UHR and HBR samples may express novel transcripts or transcripts with novel poly(A) sites, we compared the falsely predicted poly(A) sites determined by Ensembl annotation with the corresponding PolyA-seq predictions to identify true novel ones (Figure 2B, Supplementary Figure S4B). We observe a higher percentage (23.26% to 36.95%) of poly(A) sites predicted by Termin(A)$_n$tor, regardless of the species of the model, and whether they have PolyA-seq evidence support which may explain its high precision compared to the other methods. Although the raw KLEAT predictions identified a large number of poly(A) sites, only a small proportion can be verified by PolyA-seq. To maintain the precision, certain filters have to be applied to make use of KLEAT predictions, while the choice of filter may be highly library-dependent.

We compared the runtime and peak memory usage of all competing tools under the same computing environment (Table 1). All four pipelines used RNA-Bloom as the *de novo* RNA-seq assembler, which also contributed to the peak memory usage of 37.87 GB. Termin(A)$_n$tor, DeepGSR, and DeeReCT-PolyA shared the same pipeline in identifying candidate sequences, and the only difference is the NN used for sequence classification, which resulted runtime differences. DeeReCT-PolyA is the fastest tool, followed by Termin(A)$_n$tor with 10-minute (1.57%) difference. DeepGSR runs much slower than the other two NNs probably due to its implementation, which was run with Python 2.7 and older versions of Keras and Theano (the

only compatible language and packages). KLEAT is more than three times slower as the other tools because it involves both reads-to-contig and contig-to-genome alignments.

Table 1 Runtime comparison. The runtime for all four pipelines includes de novo assembly of UHR sample (two replicates), alignments, candidate poly(A) site identification and classification.

Pipeline	Wallclock Runtime (hh:mm:ss)
DeeReCT-PolyA	7:06:52
Termin(A)$_n$tor	7:11:23
DeepGSR	11:10:22
KLEAT	25:17:00

Impact of expression level on prediction accuracy

The Ensembl annotation is a repertoire of transcript annotations from multiple tissues and cell lines, so not all of the annotated poly(A) sites are expressed in the HBR and UHR samples. We quantified the abundance of all polyadenylated transcripts from Ensembl and examined how the four tools perform with respect to poly(A) sites at different expression levels (Figure 3, Supplementary Figure S5). Aside from the spikes corresponding to KLEAT raw predictions, Termin(A)$_n$tor consistently performs better than all competitors in four tiers of expression thesholds. As expected, the sensitivity of all tools increases as the expression level threshold increases, because more reads are likely to contain the poly(A) tail. The difference between Termin(A)$_n$tor and other tools increases as well when the expression cut-off increases. In fact, the metrics of Termin(A)$_n$tor mouse model on poly(A) sites expressed at > 20 TPM are similar to, or better than, DeepGSR and DeeReCT-PolyA models for both UHR and HBR samples.

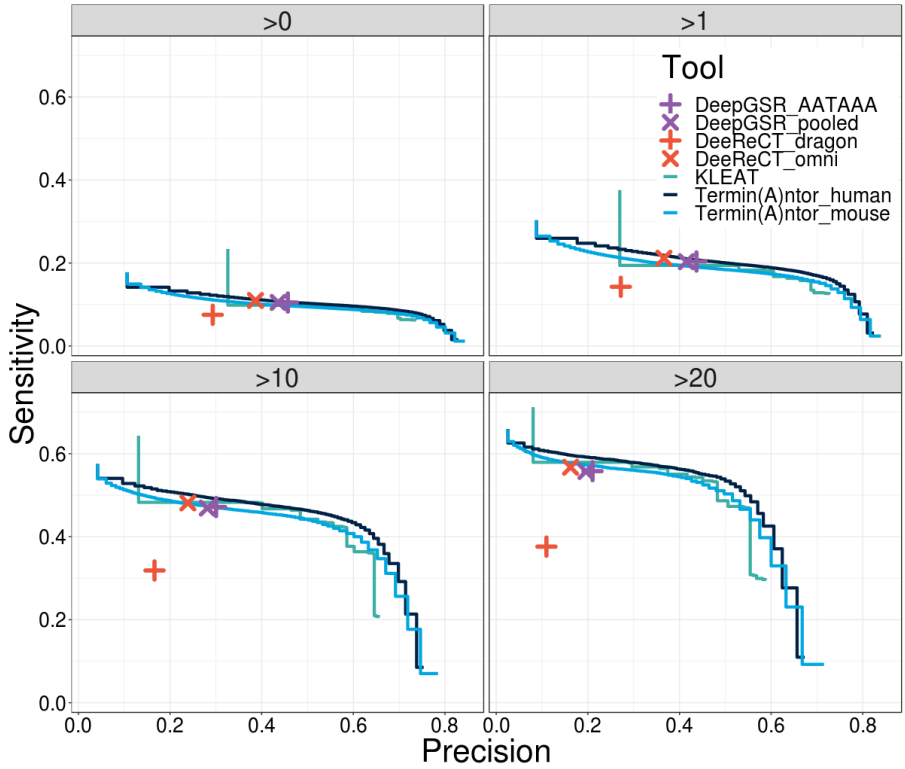

Figure 3. **Performance comparison on UHR sample with different expression level cut-offs.** Sensitivity and precision of poly(A) sites predicted by Termin(A)$_n$tor, KLEAT, DeeReCT-PolyA and DeepGSR (7 models) when compared to poly(A) sites of annotated transcripts with different expression levels. The four facet plots represent the comparison between all expressed transcripts, expressed transcript with transcript per million (TPM) > 1, > 10, and > 20.

Single nucleotide variation in PAS affects APA choices

One of the advantages of Termin(A)$_n$tor is that its capability of predicting poly(A) sites is solely based on sequence context. Consequently, poly(A) sites created or destroyed by base changes can be easily identified. We carefully examined all candidate sequences belonging to the UHR sample and identified sequences containing a different PAS hexamer from reference human genome assembly. Through comparisons between the predicted probabilities and the base variation, we discovered two cases of SNP-associated polyadenylation.

The first variant base is recorded in SNPdb as rs6484833, a C to T change at chr11:36273982 (GRCh38.p12) on the reverse strand (Figure 4A). This polymorphism creates a strong hexamer AATAAA on gene *COMMD9*, and as a result, it created an unannotated poly(A) site at chr11:36273960 on the terminus exon. In our pipeline, RNA-Bloom successfully reconstructed this novel transcript to full length with a poly(A) tail, and Termin(A)$_n$tor predicted it as a poly(A) site with 0.99 probability. This poly(A) site is also detected in PolyA-Seq of the same sample. Based on Ensembl annotation, *COMMD9* has 3 poly(A) sites associated with the same terminus exon. This site results in the second-longest 3' UTR, which harbours 3 miRNA binding sites and extensive RNA binding protein (RBP) sites for 19 RBPs (Supplementary Figure S6).

The second variant base is rs15342, a T to C change at chr15: 101070089 on gene *LRRK1* (Figure 4B). This polymorphism destroys a strong hexamer AATAAA, which is associated with an annotated poly(A) site 22 nt downstream. Figure 4B shows that 21 sequencing reads containing this polymorphism extend beyond this poly(A) site, while reads carrying the wildtype are all polyadenylated. RNA-Bloom successfully assembled both the wildtype transcript with a poly(A) tail and the variant transcript without poly(A) tail. Termin$(A)_n$tor classifed the wildtype and variant transcripts to contain poly(A) sites with a probability of 0.98 and 0.01, respectively. This polymorphism may prevent cleavage at the immediate downstream poly(A) site, leading to the alternative usage of the poly(A) site further downstream, and produces a transcript with longer 3' UTR. Similarly, extensive (16) miRNA binding sites and RBP sites for 77 RPBs are also predicted to exist on this extended 3' UTR (Supplementary Figure S7).

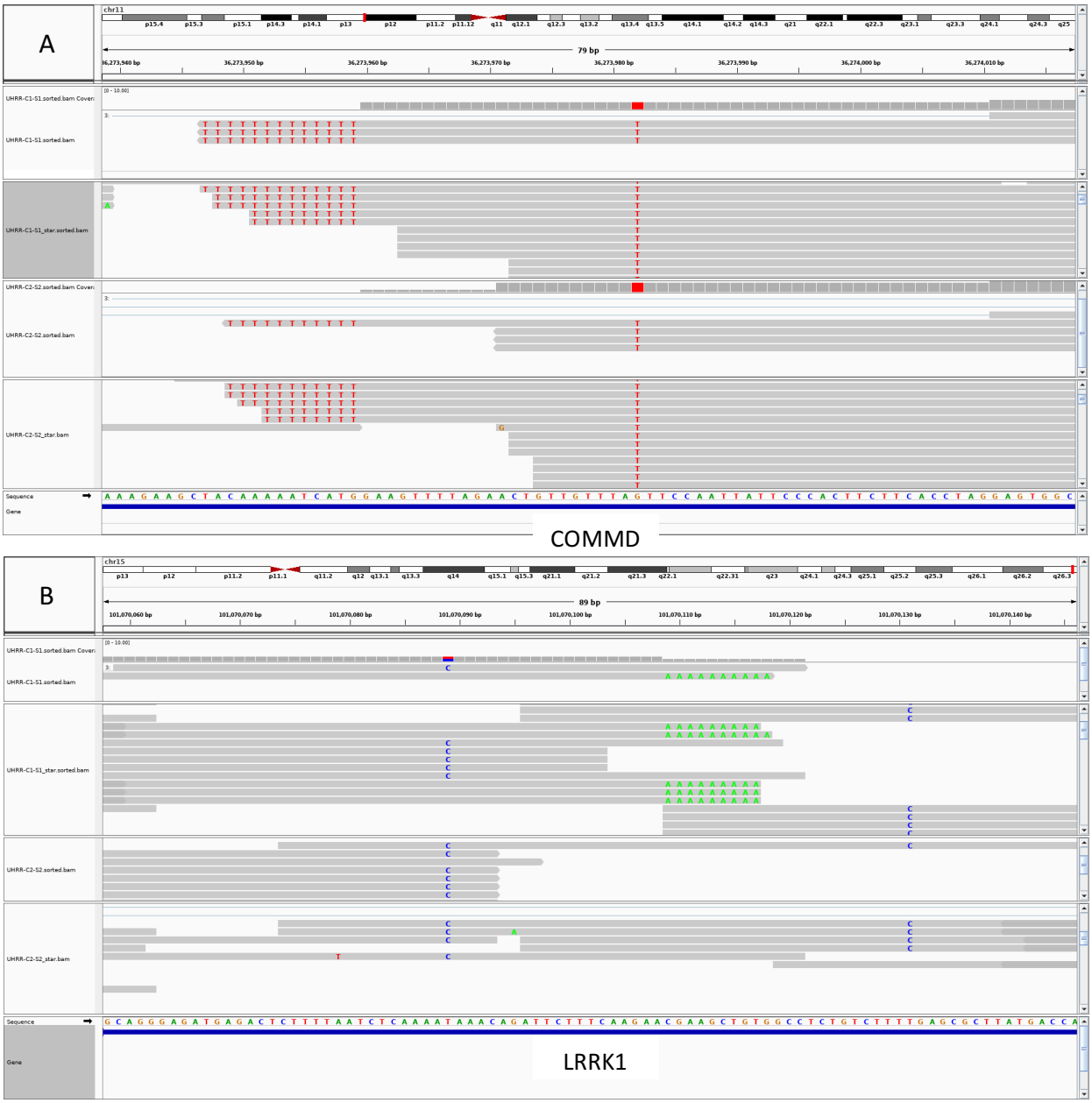

Figure 4. **Integrative Genomics Viewer (IGV) screenshot of two SNPs and the corresponding poly(A) sites**. All data shown here are based on the UHR sample. (A) SNP rs6484833 on gene *COMMD9*, (B) SNP rs15342 on gene *LRRK1*. For both panels, four tracks from top to bottom are Replicate 1 assembled transcripts ; Replicate 1 raw reads; Replicate 2 assembled transcripts; Replicate 2 raw reads.

Termin(A)$_n$tor performance on long reads

Although the current short read sequencing followed by *de novo* transcriptome assembly is the mainstream RNA-seq analysis pipeline, the advent of long read sequencing technologies provides unprecedented opportunity to study APA at the transcript isoform level. Here we applied Termin(A)$_n$tor pipeline on two sequence libraries from the human sample NA12878, namely Illumina short reads and PacBio circular consensus sequence (CCS) reads. Termin(A)$_n$tor consistently profiles three times or more poly(A) sites in PacBio library as in Illumina library under different expression level cut-offs (Figure 5). However, with Illumina library, Termin(A)$_n$tor is able to detect more poly(A) sites at a higher precision (> 70%). As the probability cut-off becomes more stringent, less positive poly(A) sites are predicted by both library, but the overlapping percentage with respect to Illumina library increases (Supplementary Figure S8). For predicted poly(A) sites with a probability > 0.9, 59.27% of the ones found in Illumina library are also found in PacBio library.

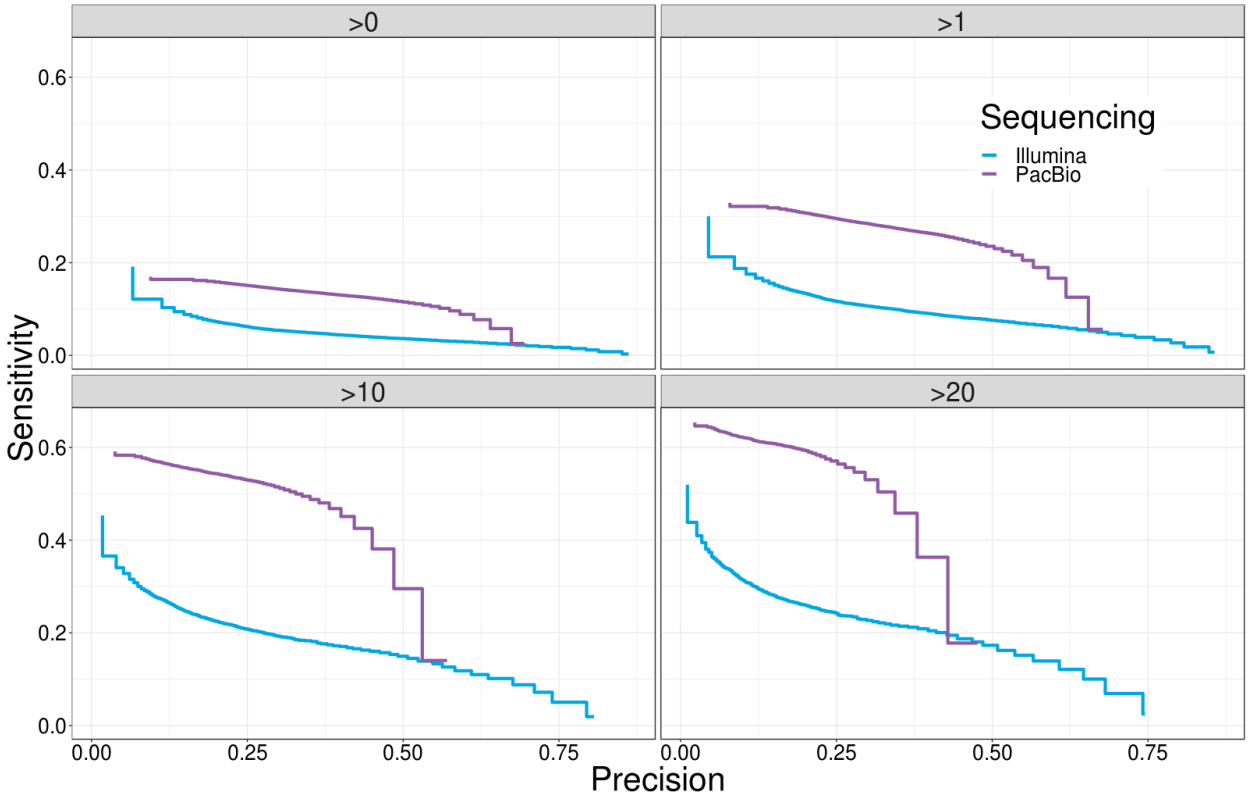

Figure 5. **Performance comparison between PacBio library and Illumina Library on NA12878 sample.** Sensitivity and precision of poly(A) sites predicted by Termin(A)$_n$tor on PacBio library and Illumina library when compared to poly(A) sites of annotated transcripts with different expression levels. The four facet plots represent the comparison between all expressed transcripts, expressed transcript with transcript

per million (TPM) > 1, > 10, and > 20.

DISCUSSION and CONCLUSIONS

Here we introduce a novel deep learning model Termin(A)$_n$tor to classify DNA sequences with or without poly(A) sites, and a profiling pipeline that starts from raw RNA-seq reads to characterize expressed poly(A) sites. The novelties of our model and contributions in this work are as follows:

1. The use of concatenated *k*-mer representations instead of raw sequence as model input. Previous applications of deep learning in omics research have shown that designing a proper representation of the raw data is critical to facilitate classification performance. Inputting additional information, namely well-studied sequence features, can also alleviate the burden of learning complex structures. In the case of polyadenylation, several *cis*-acting elements have been revealed, such as the GU-rich sequences in the downstream, UGUA elements in the upstream, and the conserved hexamer motif. Inspired by the usage of *k*-mer in genome/transcriptome assembly, we used *k*-mer representation of raw sequences to mimic the functional motifs. Since the auxiliary elements may have different lengths, multiple *k*-mer representations are jointly used to push the accuracy further.

2. The use of an embedding layer to learn the similarities between *k*-mer pairs. One-hot encoded *k*-mer representation of raw sequence is high dimensional and sparse. In our case, the length of the input is exponential to *k*. In other words, most positions in the input vector are zeros, which is computationally inefficient and cannot reflect the similarities between *k*-mers. To address this issue, we borrowed the concept of word embedding from natural language processing and used it as an interface between the input layer and the following hidden layers. Essentially, in embedding layer, one-hot encoded *k*-mers are converted into dense vectors, representing the projection of *k*-mers in a continuous vector space. As a consequence, motifs with similar functions are likely to have similar weights, and treated similarly in the following layers.

3. Separation of non-polyadenylated CSs from polyadenylated CSs. To our knowledge, Termin(A)$_n$tor is the first poly(A) site classifier that included non-polyadenylated CS label. Although most protein-coding and long-noncoding RNAs contain poly(A) tails at their 3' ends, a significant number of functional transcripts are not polyadenylated (29). Since the 3' end processing mechanism of this group is distinct from polyadenylated CS, separating it out as a new class improves the performance of the model. Sequence clustering based on Termin(A)$_n$tor's last hidden layer also groups sequences with non-polyadenylated CS and the ones with no sites together into one big cluster (Supplementary Figure S2).

Comparing to targeted 3' end sequencing technologies, RNA-seq, whether short read or long read techniques, are more accessible and widely used in transcriptome studies. In this study, we demonstrate that the Termin(A)$_n$tor pipeline is capable of identifying poly(A) sites on expressed transcripts with high precision. The failure of detecting a real and expressed poly(A)

site is mostly due to the failure of reconstructing transcripts to their 3' ends. Take UHR sample as an example, among the 58,925 expressed poly(A) sites that were not detected, more than half of them (30,181) have a TPM lower than one. Most of them have very few or no reads supporting the poly(A) tail because of sequencing bias. False positive predictions by Termin$(A)_n$tor generally occur for three reasons. The first and most common scenario are mis-assemblies due to the low sequence complexity of the 3' UTR. Second, these poly(A) sites are actually real but unannotated. Nearly 30% of the Termin$(A)_n$tor false positives according to Ensembl annotation are in fact identified by PolyA-Seq (Figure 2B, Supplementary Figure S4B). Third, mis-alignment for reference-based pipeline, which means the sequence may contain a real poly(A) site, but the genomic location is incorrect.

Theoretically, Termin$(A)_n$tor pipeline can be applied on any reconstructed transcripts to determine if its end contains a poly(A) site, or non-polyadenylated CS, or no site (3' incomplete). Aside from benchmarking its performance on traditional Illumina short read assemblies, we also explored its application on PacBio long read technology. Termin$(A)_n$tor demonstrated more than twice sensitivity in PacBio library as in Illumina library, probably because more sequences containing real poly(A) sites are fed into the NN model. PacBio CCS reads are self-corrected through consensus calling step, resulting in near- or full-length transcripts with poly(A) tails. In contrast, transcripts reconstructed from Illumina library may suffer from mis-assemblies and lowly expressed poly(A) sites may be obscured by longer transcripts. In our analysis, among the 548448 candidate sequences extracted from PacBio library, 29.71% of them contain untemplated poly(A)s, while the ratio shrunk to 7.22% out of 381934 sequences from Illumina library.

In summary, Termin$(A)_n$tor outperforms all state-of-art methods within a reasonable runtime, making it an ideal tool as a routine step in RNA-seq analysis. Termin$(A)_n$tor has demonstrated its performance on both short-read and long-read sequencing technologies. Transcriptome studies using long-read RNA-seq data will shed light on the association between poly(A) sites and transcript isoforms, a new era of APA studies. Moreover, Termin$(A)_n$tor can be used to identify novel poly(A) sites caused by base variation accurately. As has been elucidated in previous studies, novel poly(A) sites are expressed in cancer tissues, and with the help of Termin$(A)_n$tor, mechanisms and impacts of APA can be better understood. Even for species without an established poly(A) site database, their expressed poly(A) sites can be identified with a Termin$(A)_n$tor model pre-trained on a different species. The usage of Termin$(A)_n$tor, as a sequence classifier, is not limited to poly(A) site profiling, but can also be extended to facilitate gene annotation and testing the completeness of assembled transcripts. We expect Termin$(A)_n$tor to have broad applications in genome/transcriptome annotation, novel isoform identification, APA analysis and gene regulation studies.

METHODS

Data sets

In this work, we built two data sets (human and mouse) for model training, cross-validation,

and cross-species performance demonstration. Each data set is composed of sequences with equal lengths coming from three classes, poly(A) site, non-polyadenylated CS, and no sites (Table 1). PolyA_DB version3.1 was used as our primary poly(A) site library (11). We first mapped poly(A) sites recorded in PolyA_DB3 to Ensembl annotation release 94 using `bedtools closest` (v2.27.1) and selected the most compatible transcript for each site. Poly(A) sites supported by less than 1 Reads Per Million (RPM) or located more than 1000 nt downstream of any annotated transcript were considered unreliable and were discarded.

To avoid potential poly(A) sites from contaminating the other two classes, we expanded our poly(A) site library by incorporating three additional PolyA databases, including APADB v2, APASdb, and PolyASite. Again, we used `bedtools closest` to assign poly(A) sites in the compiled poly(A) library to Ensembl transcripts. Transcripts with no poly(A) sites assigned were considered as cleaved but not polyadenylated and their 3' ends constituted the non-polyadenylated CS set.

Pseudo sites comprises 5 types of genomic locations: exonic, intronic, intergenic, 5' and 3' untranslated regions, which are randomly selected and at least 100 nt away from any annotated 3' ends, including Ensembl annotation and compiled poly(A) library.

Our data sets are imbalanced due to the limited number of non-polyadenylated CS. However, to fully exploit the potential of deep NNs, we used all non-redundant poly(A) sites and CS generated as above descriptions and deliberately matched the number of pseudo sites to poly(A) sites. As a consequence, we used both accuracy and the F-measure, the harmonic mean of precision and recall, as assessment metrics during model training and benchmarking.

Table 2. Size and origin of data sets for model training, testing, and validation

Species	**Poly(A) site**	**CS**	**Pseudo site**
H. sapiens	52,457	27,595	52,460
M. musculus	75,661	37,028	75,660

Model training

The NN model is implemented in Python 3.6.6, using Keras library 2.2.4 with TensorFlow 1.11.0 as the backend (30,31). To evaluate the performance of the model with different architectures and hyperparameter combinations, we used stratified 5-fold cross validation on the whole data set to monitor the overall accuracy of three labels, and accuracy, sensitivity, and specificity on poly(A) label. All performance metrics on the training sets reported in this study are the mean and standard deviation of 5 validation processes. In each process, 80% of data is used for training and 20% is used for validation. Next, we trained the model using the entire training set and used it for benchmarking the performance on experimental data.

Model architecture and hyperparameter tuning

Searching for the optimal hyperparameters for a deep NN has always been a tedious task, as many of them are dependent on each other. We first built the basic frame with one input layer, one embedding layer, and one output classification layer, and then fine-tune hyperparameters related to architecture and training using grid search as listed in Table 3. For all models, initial model weights were randomly drawn from Glorot uniform initializer (32). The weights of the three neurons in the output layer are computed via softmax activation function, representing the probability distribution of the three labels and they always add up to one. In order to avoid over-fitting, we applied early stopping to each model. As an iterative method, the validation loss is monitored at each epoch and when it stays the same or increases for the next ten epochs, the training process will be stopped. Without early stopping, too many training epochs will lead to nearly 100% accuracy on training set, yet very low accuracy on validation and test set.

Table 3. Model hyperparameter tuning. Bolded parameters are the selected optimal values.

Parameter type	Parameter	Search space
Training	Optimizer	SGD, **Adam**, Adagrad, RMSprop
	Learning rate	0.1, 0.01, **0.001**, 0.0001
	Loss function	**Categorical_cross_entropy**, mean_squared_error
	K-mer representation	{1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10}, {11}, {12}, {4, 5}, {4, 6}, {4, 8}, {4, 10}, {4, 11}, {4, 5, 6}, {4, 6, 8}, {4, 5, 6, 10}, **{4, 6, 8, 10},** {4, 5, 6, 8, 10}
	Batch size	32, **64**, 128, 256
Architecture	Number of layers	1, 2, 3, **4**, 5
	Number of nodes for each layer	4, 16, **32**, **64**, **128**, 256, **512**, 1024 Refer to Figure 6 for details
	Regularization	Dropout: **0**, 0.1, 0.2, 0.5
	Activation function for hidden layers	Tanh, **ReLU**
	Activation function for output layer	**Softmax**, Sigmoid

The final architecture of Termin(A)$_n$tor is depicted in Figure 6 and the selected optimal hyperparameters are in bold font in Table 3. The first step of Termin(A)$_n$tor is to concatenate all *k*-mer representations of the raw DNA sequence. Each *k*-mer in the one-hot encoded raw sequence is represented as a vector of length l as computed in Formula (1), which is exponential to *k*.

$$l = \sum_{k=4}^{10} 4^{k}, \ k = \{4, 6, 8, 10\} \tag{1}$$

Then an embedding layer with 128 nodes is used to connect the input layer and hidden layers. The next four layers are all fully connected layers and the activation function between each two

layers is rectified linear unit (ReLU). Since dropout regularization did not achieve better performance, they were removed from the final model.

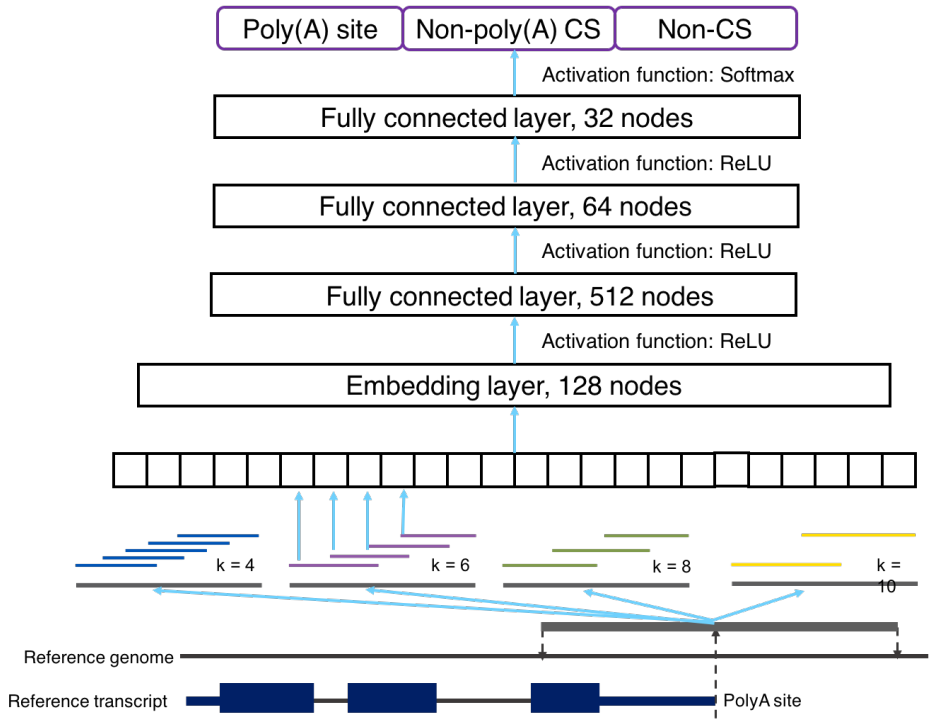

Figure 6. Architecture of Termin(A)$_n$tor

Experimental data

The raw RNA-seq reads for the two MAQC reference human samples, HBR and UHR, were downloaded from Illumina BaseSpace, and each sample has two replicates (https://basespace.illumina.com/projects/3777775/samples) (33). Raw RNA-seq reads Each of the 4 libraries was prepared with Illumina TruSeq Stranded kits using poly(A) selection and sequenced as 75bp paired-end reads. The Illumina and PacBio reads for NA12878 were obtained from an earlier study under accession number SRR1153470 and SRR1163655, respectively (34). The Illumina reads are poly(A) selected and sequenced as 75bp stranded paired-end reads.

Prediction pipeline

Here we propose a Poly(A) site prediction pipeline based on Termin(A)$_n$tor. The pipeline starts with RNA-Bloom, a fast and memory-efficient *de novo* transcriptome assembler (35). RNA-Bloom is run with the option `-stratum 01` that allows extension of all fragments regardless of its coverage, and the option `--polya` that prioritizes the assembly of transcripts with poly(A) tails. The false positive rate (FPR) of Bloom filters in the program is set to 0.005. Then, all assembled transcripts are aligned to the reference genome hg38 and compared to Ensembl annotation release 94 to identify the ones that miss poly(A) tail but do end in the 3' UTR of

annotated transcripts. These transcripts, together with the ones end with a stretch of untemplated As, are considered as having potential poly(A) sites. Similar to sequences in the training data set, a test sequence can also be divided into three parts, 100nt upstream sequence extracted from the assembled transcript, a potential poly(A) site, and 100nt downstream genomic sequences from the reference. Finally, these candidate sequences are fed into pre-trained Termin$(A)_n$tor for classification.

Evaluation of prediction pipeline

Poly(A) sites are heterogeneous, so they are often observed as a group of neighbouring sites rather than a precise genomic location (6). Predicted poly(A) sites within 30nt were clustered and the one with the highest probability was chosen. Ensembl annotated poly(A) sites were also clustered in the same way, while the one located in the median position was chosen as the representative. Then, the predicted list was compared with the annotated list to compute the hits and misses. In our evaluation system, true positive poly(A) site is a predicted poly(A) site that lies within ± 30 nt of an annotated site; false positive site is a predicted poly(A) site that does not lie within ± 30 nt of any annotated site; false negative means an annotated site does not have any predicted site lie within ± 30 nt; true negative means there is no real nor predicted poly(A) site in a certain location, which cannot be monitored. Since true negative is missing in the evaluation system, we used sensitivity and precision as performance metrics for all experimental data tests.

Tool comparison

We applied the proposed prediction pipeline on experimental data and compared the performance with state-of-the-art methods. All experiments were run on Centos 6.7 system with 12 Intel Xeon E5-2650 CPUs and 84 GB memory.

KLEAT pipeline was chosen as a representative of read-evidence based approach. The auxiliary files KLEAT needed include: 1) RNA-seq assemblies from RNA-Bloom, 2) read to contig alignment file generated by BWA-MEM 0.7.17-r1188 (36), and 3) contig to genome alignment file generated by GMAP version 2017-01-14 (37). Since KLEAT only works on hg19 assemblies, all predicted poly(A) sites were converted into hg38 genomic coordinates by UCSC liftOver (38). KLEAT outputs all possible poly(A) sites and their supporting evidence, such as the length of poly(A) tail in reconstructed transcripts, the number of reads containing poly(A) tail, and the length of poly(A) tail of reads that can be mapped to a reconstructed transcript. We applied filters to the raw predictions with different combinations of three types of evidences and computed the Pareto frontier for each plot.

Two emerging deep learning models, DeepGSR and DeeReCT-PolyA, were also included in the comparison. Similar to Termin$(A)_n$tor, these two tools are simply NN models expecting fixed length sequences as input. Starting from all candidate sequences processed from our prediction pipeline, we selected only the ones containing PAS, adjusted the sequence lengths as required, and fed into these two models for classification. For DeepGSR, we used two pre-trained models, one is trained on only AATAAA signal and the other is trained on all signals. DeeReCT-PolyA also

published two pre-trained models based on two data sets, dragon and omni, both of which were included in the comparison.

LIST OF ABBREVIATIONS

3' untranslated region	3'UTR
Alternative polyadenylation	APA
Circular consensus sequence	CCS
Cleavage site	CS
Human Brain Reference	HBR
Neural network	NN
Pacific Bioscience	PacBio
Polyadenylation	poly(A)
Poly(A) signal	PAS
RNA Binding Protein	RBP
Reads Per Million	RPM
Rectified linear unit	ReLU
Single Molecule, Real-Time	SMRT
Single nucleotide polymorphism	SNP
Transcripts Per Million	TPM
Universal Human Reference	UHR

DECLARATIONS

Ethics approval and consent to participate

Consent for publication

Availability of data and material

Termin$(A)_n$tor is implemented using Python 3.6 with Keras package. The software package is freely available under GNU General Public Licence. The source code and manual are available at the GitHub repository https://github.com/bcgsc/TerminAntor. The human and mouse data sets generated in this study are included in Additional file 2 and 3. Previously published data used in this work are described in the Methods section, and are available from several public repositories.

Competing interests

The authors decare that they have no competing interests.

Funding

This work is funded by XXXXXX (ID1324).

Authors' contributions

IB and CY conceived and designed the study. CY designed and implemented the software with the help of CL, KMN and RLW. KMN provided the additional help in the pipeline design. CY drafted the manuscript and all the other authors were involved in revision and provided helpful feedback. All authors read and approved the final manuscript.

Acknowledgements

REFERENCE

1. Edmonds M, Vaughan MH, Nakazato H. Polyadenylic acid sequences in the heterogeneous nuclear RNA and rapidly-labeled polyribosomal RNA of HeLa cells: possible evidence for a precursor relationship. Proc Natl Acad Sci U S A. 1971;68(6):1336–40.
2. Neve J, Patel R, Wang Z, Louey A, Furger AF. Cleavage and polyadenylation: Ending the message expands gene regulation. RNA Biol. 2017;14(7):865–90.
3. Barrett LW, Fletcher S, Wilton SD. Regulation of eukaryotic gene expression by the untranslated gene regions and other non-coding elements. Vol. 69, Cellular and Molecular Life Sciences. 2012. p. 3613–34.
4. Shi Y, Chan S, Martinez-Santibañez G. An up-close look at the pre-mRNA 3'-end processing complex. RNA Biol [Internet]. 2009;6(5):522–5. Available from: http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&id=19713761&retmode=ref&cmd=prlinks%5Cnpapers2://publication/uuid/7226CF03-DF4D-4C38-B194-31F4CAE25111
5. Zhang H, Lee JY, Tian B. Biased alternative polyadenylation in human tissues. Genome Biol [Internet]. 2005;6(12):R100. Available from: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=PubMed&dopt=Citation&list_uids=16356263%5Cnhttp://www.ncbi.nlm.nih.gov/pubmed/16356263%5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1414089
6. Tian B, Hu J, Zhang H, Lutz CS. A large-scale analysis of mRNA polyadenylation of human and mouse genes. Nucleic Acids Res. 2005;33(1):201–12.
7. Lianoglou S, Garg V, Yang JL, Leslie CS, Mayr C. Ubiquitously transcribed genes use alternative polyadenylation to achieve tissue-specific expression. Genes Dev. 2013;27(21):2380–96.
8. Lembo A, Di Cunto F, Provero P. Shortening of 3'UTRs correlates with poor prognosis in breast and lung cancer. PLoS One. 2012;7(2).
9. Li L, Wang D, Xue M, Mi X, Liang Y, Wang P. 3'UTR shortening identifies high-risk cancers with targeted dysregulation of the ceRNA network. Sci Rep [Internet]. 2015;4(1):5406. Available from: http://www.nature.com/articles/srep05406
10. Aken BL, Ayling S, Barrell D, Clarke L, Curwen V, Fairley S, et al. The Ensembl gene annotation system. Database [Internet]. 2016;2016:baw093. Available from: https://academic.oup.com/database/article-lookup/doi/10.1093/database/baw093
11. Wang R, Nambiar R, Zheng D, Tian B. PolyA-DB 3 catalogs cleavage and polyadenylation sites identified by deep sequencing in multiple genomes. Nucleic Acids Res. 2018;46(D1):D315–9.

12. Liu D, Brockman JM, Dass B, Hutchins LN, Singh P, McCarrey JR, et al. Systematic variation in mRNA 3'-processing signals during mouse spermatogenesis. Nucleic Acids Res. 2007;35(1):234–46.
13. Mayr C. Evolution and Biological Roles of Alternative 3'UTRs. Vol. 26, Trends in Cell Biology. 2016. p. 227–37.
14. Shepard PJ, Choi E-A, Lu J, Flanagan LA, Hertel KJ, Shi Y. Complex and dynamic landscape of RNA polyadenylation revealed by PAS-Seq. RNA [Internet]. 2011;17(4):761–72. Available from: http://rnajournal.cshlp.org/cgi/doi/10.1261/rna.2581711
15. Harrison PF, Powell DR, Clancy JL, Preiss T, Boag PR, Traven ANA, et al. PAT-seq : a method to study the integration of 3' -UTR dynamics with gene expression in the eukaryotic transcriptome. Rna [Internet]. 2015;21(8):1502–10. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4509939&tool=pmcentrez&rendertype=abstract%5Cnhttp://rnajournal.cshlp.org/cgi/doi/10.1261/rna.048355.114
16. Chang H, Lim J, Ha M, Kim VN. TAIL-seq: Genome-wide determination of poly(A) tail length and 3' end modifications. Mol Cell. 2014;53(6):1044–52.
17. Derti A, Garrett-Engele P, MacIsaac KD, Stevens RC, Sriram S, Chen R, et al. A quantitative atlas of polyadenylation in five mammals. Genome Res. 2012;22(6):1173–83.
18. BIROL I, RAYMOND A, CHIU R, NIP KM, JACKMAN SD, KREITZMAN M, et al. Kleat: Cleavage Site Analysis of Transcriptomes. Biocomput 2015 [Internet]. 2014;347–58. Available from: http://www.worldscientific.com/doi/abs/10.1142/9789814644730_0034
19. Bonfert T, Friedel CC. Prediction of poly(A) sites by poly(A) read mapping. PLoS One. 2017;12(1).
20. Xia Z, Donehower LA, Cooper TA, Neilson JR, Wheeler DA, Wagner EJ, et al. Dynamic analyses of alternative polyadenylation from RNA-seq reveal a 3'-UTR landscape across seven tumour types. Nat Commun [Internet]. 2014;5:5274. Available from: http://www.nature.com/doifinder/10.1038/ncomms6274
21. Ye C, Long Y, Ji G, Li QQ, Wu X. APAtrap: Identification and quantification of alternative polyadenylation sites from RNA-seq data. Bioinformatics. 2018;34(11):1841–9.
22. Ha KCH, Blencowe BJ, Morris Q. QAPA: A new method for the systematic analysis of alternative polyadenylation from RNA-seq data. Genome Biol. 2018;19(1).
23. Kalkatawi M, Magana-Mora A, Jankovic B, Bajic VB. DeepGSR: an optimized deep-learning structure for the recognition of genomic signals and regions. Bioinformatics [Internet]. 2018; Available from: https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/bty752/5089227
24. Xia Z, Li Y, Zhang B, Li Z, Hu Y, Chen W, et al. DeeReCT-PolyA: a robust and generic deep learning method for PAS identification. Bioinformatics [Internet]. 2018 Nov 30;bty991-bty991. Available from: http://dx.doi.org/10.1093/bioinformatics/bty991
25. Chang TH, Wu LC, Chen YT, Huang H Da, Liu BJ, Cheng KF, et al. Characterization and prediction of mRNA polyadenylation sites in human genes. Med Biol Eng Comput. 2011;49(4):463–72.
26. Xie B, Jankovic BR, Bajic VB, Song L, Gao X. Poly(A) motif prediction using spectral latent features from human DNA sequences. In: Bioinformatics. 2013.
27. Venkataraman K, Brown KM, Gilmartin GM. Analysis of a noncanonical poly(A) site reveals a tripartite mechanism for vertebrate poly(A) site recognition. Genes Dev.

2005;19(11):1315–27.
28. Nunes NM, Li W, Tian B, Furger A. A functional human Poly(A) site requires only a potent DSE and an A-rich upstream sequence. EMBO J. 2010;29(9):1523–36.
29. Yang L, Duff MO, Graveley BR, Carmichael GG, Chen LL. Genomewide characterization of non-polyadenylated RNAs. Genome Biol. 2011;12(2).
30. Chollet F. Keras: Deep Learning for humans [Internet]. Github. 2015. Available from: https://github.com/keras-team/keras
31. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. TensorFlow : A System for Large-Scale Machine Learning This paper is included in the Proceedings of the TensorFlow : A system for large-scale machine learning. Proc 12th USENIX Conf Oper Syst Des Implement. 2016;272–83.
32. Glorot X, Statistics YBBT-P of the TIC on AI and. Understanding the difficulty of training deep feedforward neural networks [Internet]. Teh YW, Titterington M, editors. PMLR ; 2010. p. 249–56. Available from: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf
33. Shi L, Reid LH, Jones WD, Shippy R, Warrington JA, Baker SC, et al. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. Nat Biotechnol. 2006;24(9):1151–61.
34. Tilgner H, Grubert F, Sharon D, Snyder MP. Defining a personal, allele-specific, and single-molecule long-read transcriptome. Proc Natl Acad Sci. 2014;111(27):9869–74.
35. Nip KM, Chiu R, Yang C, Chu J, Mohamadi H, Warren RL, et al. RNA-Bloom provides lightweight reference-free transcriptome assembly for single cells. bioRxiv [Internet]. 2019 Jan 1;701607. Available from: http://biorxiv.org/content/early/2019/07/14/701607.abstract
36. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv Prepr arXiv [Internet]. 2013;0(0):3. Available from: http://arxiv.org/abs/1303.3997
37. Wu TD, Watanabe CK. GMAP: A genomic mapping and alignment program for mRNA and EST sequences. Bioinformatics. 2005;21(9):1859–75.
38. Hinrichs AS, Karolchik D, Baertsch R, Barber GP, Bejerano G, Clawson H, et al. The UCSC Genome Browser Database: update 2006. Nucleic Acids Res [Internet]. 2006;34(Database issue):D590–8. Available from: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1347506&tool=pmcentrez&rendertype=abstract