Terminitor: Cleavage Site Prediction Using Deep Learning Models

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

[1] Canada's Michael Smith Genome Sciences Centre, BC Cancer, Vancouver, V5Z 4S6, Canada

[2] University of British Columbia, Vancouver, V6T 1Z4, Canada

* Corresponding author

E-mail: ibirol@bcgsc.ca (Birol I)

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 known to be influenced by library preparation artefacts or manually selected features. Moreover, recently published machine learning methods have only been tested on pre-constructed datasets, thus lacking validation on experimental data. Here we present Terminitor, the first deep neural network-based profiling pipeline to make predictions from RNA-seq data. We show how Terminitor outperforms competing tools in sensitivity and precision on experimental transcriptome sequencing data, and demonstrate its use with data from short- and long-read sequencing technologies. For species without a good reference transcriptome annotation, Terminitor is still able to pass on the information learnt from a related species and make reasonable predictions. We used Terminitor to showcase how single nucleotide variations can create or destroy polyadenylated cleavage sites in human RNA-seq samples.

Author Summary

3’ cleavage and polyadenylation of pre-mRNA is part of RNA maturation process. One gene can be cleaved at different positions at its 3’ end, namely alternatively polyadenylation, thus identifying the correct polyadenylated cleavage site (poly(A) CS) is essential to unveil its role in gene regulation under different physiological and pathological conditions. The current poly(A) CS prediction tools are either heavily influenced by RNA-Seq library preparation artefacts or have only been designed and tested on *ad hoc* datasets, lacking association with real world applications. In this study, we present a deep learning model, Terminitor, that predicts the probability of a nucleotide sequence containing a poly(A) CS, and validated its performance on human and mouse data. Along with the model, we propose a poly(A) CS profiling pipeline for RNA-seq data. We benchmarked our pipeline against competing tools and achieved higher sensitivity and precision in experimental data. The usage of Terminitor is not limited to genome and transcriptome annotation and we expect it to facilitate the identification of novel isoforms, improve the accuracy of transcript quantification and differential expression analysis, and contribute to the repertoire of reference transcriptome annotation.

Introduction

Since polyadenylated (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 [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. This process is facilitated by several *cis*-acting RNA elements, represented by the poly(A) signal (PAS), conserved hexamer motifs with AATAAA being the most strongest, located 10-30 nt upstream of the cleavage site (CS) [6]. Because most nascent RNA in eukaryotes have more than one possible 3' end CS, mRNAs with different 3' ends are formed following these two coupled processes [7]. This phenomenon is also known as alternative polyadenylation (APA), which results in transcripts with different 3' untranslated regions (UTR). In recent years, advances in DNA/RNA sequencing technology allowed detailed study of APA modulation under different physiological and pathological conditions, and shed light on its implications in some diseases, especially cancer [5,8–10].

The importance of profiling 3' UTR cleavage

APA, together with alternative splicing, has considerable impact on the modulation of gene expression and contributes to transcriptome complexity. As of date, the catalogue of profiled poly(A) CS, especially cancer-specific ones, is far from complete [11]. Ensembl annotation version GRCh38.94 records 63,620 poly(A) CS for 19,907 protein coding genes in human, and among them, 73.13% genes can produce two or more APA isoforms (S1 Fig) [12]. The most recent poly(A) database, PolyA_DB3, compiled 24 human samples and cataloged 108,042 poly(A) CS for 20,998 genes [13]. The substantial improvement in 3' end annotation compared to Ensembl suggests the

66 sample-specific usage of poly(A) CS, highlighting the incompleteness of the current annotation.
67 Moreover, usage of these poly(A) CS is dynamically regulated under different developmental
68 stages and pathological changes, as elucidated in previous studies [5,8,14,15]. Profiling poly(A)
69 CS with respect to biological conditions serves as the first step towards deciphering the
70 underlying mechanism of APA-mediated gene regulation. Hence, it is desirable to incorporate a
71 fast and robust poly(A) CS characterization tool into standard transcriptome analysis pipelines.

72 **Current methods and their limitations**

73 In the past decades, continuous efforts have been made to annotate 3' ends in the genome
74 and predict poly(A) CS in the transcriptome, using methods designed for two major sequencing
75 protocols: 3' end targeted sequencing and poly(A)-selected RNA-seq. The former is a specialized
76 high throughput sequencing technique that is developed to overcome the low sequence
77 complexity at the 3' end of transcripts. These methods, including TAIL-Seq, PolyA-Seq, PAS-seq,
78 and PAT-seq [16–19], only interrogate the 3' ends of transcripts by primers, followed by
79 sequencing and bespoke analysis pipelines. Compared to methods that use RNA-seq, a critical
80 advantage of these protocols is their high sensitivity in detecting poly(A) CS from lowly expressed
81 transcripts. With the help of 3' end sequencing technologies, several poly(A) CS databases have
82 been established [13,20–22]. Though powerful, they require specialized library preparation,
83 which can be costly and laborious, and their utility is restricted to poly(A) detection. As a result,
84 they have not yet been widely adopted in genetic research.

85 In contrast, RNA-seq is the established sequencing method of choice for a wide range of
86 transcriptomics projects. It can also be used for poly(A) CS profiling on a per-sample basis, as well
87 as for systematic APA analysis of large cohorts of individuals, including retrospective studies.

88 Poly(A) CS prediction and annotation tools designed for RNA-seq have demonstrated that with
89 suitably redundant coverage (read coverage ≥ 2), RNA-seq data type contains sufficient
90 information for the identification of poly(A) tails, and thus permits the discovery of novel poly(A)
91 CS [23,24]. Generally speaking, these tools can be classified into three approaches: read-evidence
92 based, expression level based, and machine learning based.

93 Read-evidence based approaches, such as KLEAT [23] and ContextMap2 [24], search for
94 untemplated adenosines (As) in aligned sequences as evidence to determine the location of
95 poly(A) CS. Although these tools are able to discover novel poly(A) CS with high resolution, they
96 may lose sensitivity in low coverage sequencing libraries and when transcript expression is low.

97 For approaches based on expression levels, poly(A) CS prediction is a by-product of APA
98 analysis. Tools adopting this approach, which include DaPars, APAtrap, and QAPA [25–27], all rely
99 on differential expression in the 3' UTRs of matched samples. However, due to the low sequence
100 complexity of 3' UTRs and the resulting sequencing data artefacts, read coverage in these regions
101 may not reflect true expression levels, impacting differential expression analysis and associated
102 statistical tests. Consequently, these methods may fail to detect real but lowly represented APA
103 events. Moreover, they only report poly(A) CS involved in significant APA events through
104 comparative analyses, and thus these tools are not capable of profiling poly(A) CS in given
105 samples.

106 Another approach to address the prediction problem is through the use of deep learning
107 methods, and several such models have been proposed recently, such as DeeReCT-PolyA [28],
108 DeepGSR [29], and DeepPASTA [30]. For these tools, hidden features are learnt and modeled
109 directly from input nucleotide sequences. Using this approach, DeeReCT-PolyA and DeepGSR

110 have been reported to show around 90% accuracy on test sequences [28,29]. Combining both
111 sequence and RNA secondary structure information, DeepPASTA, was reported to have an area
112 under the curve score over 93% in predicting poly(A) CS. However, these tools were only bench-
113 marked on pre-constructed datasets composed of positive and negative genomic sequences, and
114 none have been applied to real experimental transcriptome data for prediction. Another
115 limitation of DeeReCT-PolyA and DeepGSR is that they were only trained to distinguish whether
116 a PAS is real. However, it is known that a small fraction (yet a substantial number) of functional
117 human poly(A) CS does not require a PAS [32,33]. At present, these approaches overlook poly(A)
118 CS with noncanonical *cis*-acting elements, which would likely negatively impact their
119 performance on experimental data.

120 **Terminitor**

121 Despite the availability of various tools, comprehensively and efficiently profiling poly(A) CS in
122 RNA-seq samples remains challenging. In this study, we present Terminitor, a deep neural
123 network (NN) model for fast and accurate poly(A) CS recognition, independent of the existence
124 of a PAS. Terminitor takes fixed length sequences as input and performs a three-label
125 classification to determine whether the sequence contains a poly(A) CS, a non-polyadenylated
126 CS, or non-CS. Coupled with Terminitor, we also propose a profiling pipeline that generates
127 prediction of poly(A) CS using raw RNA-seq data as input. The performance of this pipeline was
128 cross-validated on two datasets of sequences (describing human and mouse transcriptomes) to
129 select the optimal input sequence length and model hyperparameters. We have benchmarked
130 our pipeline against competing tools on experimental paired-end RNA-seq libraries generated
131 from the Illumina (San Diego, CA) TruSeq paired-end and Pacific Biosciences (Menlo Park, CA)

132 Single Molecule, Real-Time (SMRT) Sequencing platforms.

133 Results

134 Performance of Terminitor using different sequence lengths

135 Terminitor is trained on experimentally validated annotated nucleotide sequences containing
136 three classes of sites: (1) poly(A) CS, (2) non-poly(A) CS, and (3) non-CS. Using cross-validated
137 optimal hyperparameters, it computes the probability of a test sequence belonging to one of the
138 three classes.

139 To train, validate, and test the model, we collated two reference datasets describing human
140 and mouse transcriptomes, based on poly(A) CS databases and Ensembl annotations. These
141 datasets are composed of labeled sequences of uniform length with three parts: upstream
142 sequence, a site from one of the three classes, and downstream sequence. Upstream sequences
143 are from corresponding transcripts for the first two classes and from genomic regions for the
144 third class, while downstream sequences are genomic, and the transition from upstream
145 sequence to downstream sequence is marked as the cleavage site or lack thereof (S2 Fig). Since
146 there is no consensus from previous work on how the input sequence length influences
147 prediction accuracy, we built models with different lengths to evaluate its effect on performance.
148 Eight upstream lengths (200, 160, 120, 100, 80, 60, 40 and 20 nt) and five downstream lengths
149 (200, 100, 40, 20 and 0 nt) were chosen to create a total of 40 models, which were trained to
150 explore the impact of sequence length on poly(A) CS prediction (Fig 1). In general, all models
151 show clear separation between poly(A) CS and the other two labels (S3 Fig). The accuracy of all
152 40 models is above 87%, including the one trained with only 20 nt long upstream sequence. As
153 expected, models trained with sequences containing both upstream and downstream sequences

154 perform better than the ones trained with upstream sequences alone (0 nt downstream
155 sequence length). However, given the same upstream sequences, models trained with
156 downstream sequences longer than 20 nt perform comparably to each other. The 28 models
157 trained with upstream sequence lengths in {40, 60, 80, 100, 120, 160, 200} and downstream
158 sequence lengths in {20, 40, 100, 200} have an average accuracy of 94.50% ± 0.0015. More
159 interestingly, longer downstream sequences do not necessarily lead to better performance.
160 Models trained with 200 nt downstream sequences have slightly lower accuracy than the ones
161 trained with shorter lengths, suggesting that some polyadenylation recognition patterns maybe
162 obscured by sequence motifs further downstream.

163 **Fig 1. Model performance on human reference dataset.** Each facet represents one
164 downstream sequence length, and the x-axis of each facet plot shows the up-stream sequence
165 length. Each data point is the mean of 5-fold cross validation and the error bar is the standard
166 deviation.

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

174 **Tool comparison on experimental data**

175 In order to demonstrate the application of Terminitor on poly(A) CS profiling, we designed a
176 prediction pipeline and applied it on two human RNA-seq samples, Human Brain Reference RNA

177 (HBR) and Universal Human Reference RNA (UHR), to predict poly(A) CS from reconstructed
178 transcripts. As done previously, we trained Terminitor with 40 length combinations, applied all
179 40 models on UHR sample, and computed the Pareto frontier based on sensitivity and precision.
180 The sensitivity of Terminitor is positively correlated with upstream sequence length, while there
181 is no clear trend between the downstream sequence length and the performance metrics (S4 Fig).
182 For demonstration purposes, we chose ± 100 nt model for performance benchmarking and
183 comparison.

184 Profiling poly(A) CS for species without a good reference annotation can be challenging. Thus
185 it is interesting to test whether the information learnt from one species is transferable to another.
186 In addition to the model built on the above-mentioned human reference dataset, we also
187 generated a model that was separately trained on the mouse dataset. We applied these two
188 models on the UHR and HBR samples and compared the performance with KLEAT, DeepPASTA,
189 DeeReCT-PolyA and DeepGSR (Fig 2A, S5A Fig). Poly(A) CS predicted by these methods were
190 compared against Ensembl annotation to calculate the sensitivity and precision. Note that the
191 denominator in sensitivity calculations represents all Ensembl annotations, yet for a given sample
192 not all transcripts would be expressed. Without a *priori* information on gene and isoform
193 expression, this is the most appropriate solution at it penalizes all methods the same way,
194 preserving their relative performance for fair comparisons.

195 **Fig 2. Performance comparison on the UHR sample. A.** Sensitivity and specificity of poly(A)
196 CS predictions from Terminitor, KLEAT, DeepPASTA, DeeReCT-PolyA and DeepGSR on Ensembl
197 annotated poly(A) CS. The two pre-trained models of DeepGSR are the one with sequences
198 containing only the strongest PAS hexamer AATAAA, and the one with sequences containing 16
199 hexamer motifs pooled together. The two pre-trained models of DeeReCT-PolyA are derived from
200 the dragon dataset and omni dataset as described (Xia et al., 2019). Two pre-trained Terminitor
201 models are derived from the human and mouse datasets. The navy/blue crosses on Terminitor

202 human/mouse model represent probability = 0.5 cut-off, respectively. **B.** Percentage of poly(A)
203 CS that are missing from Ensembl annotation are supported/not supported by PolyA-Seq. We
204 used probability = 0.5 cut-off for Terminitor and DeepPASTA predictions.

205 We also note that DeeReCT-PolyA and DeepGSR are binary classifiers, while Terminitor
206 outputs a probability value that can be tuned in favour of either sensitivity or precision.
207 Nevertheless, Terminitor trained on human sequences consistently outperforms all competing
208 tools in both metrics, except for one data point, which corresponds to the raw, unfiltered KLEAT
209 predictions. However, the precision of raw KLEAT predictions is arguably too low (35.90% and
210 25.52% for UHR and HBR samples, respectively) to be used for high-confidence profiling.
211 Although KLEAT performs comparably to Terminitor on UHR sample, its performance is
212 surprisingly low on HBR sample regardless of what read evidence is used to filter its raw
213 predictions. It is also worth mentioning that the four models built by DeeReCT-PolyA and
214 DeepGSR all have a precision lower than 50%, while precision of the Terminitor human model (at
215 probability = 0.5 cut-off) is 55.59% and 57.72% for UHR and HBR samples, respectively.

216 Since a given sample may express novel transcripts or transcripts with novel poly(A) CS, we
217 compared the falsely predicted poly(A) CS determined by Ensembl annotation with the
218 corresponding PolyA-seq predictions to identify putative novel events with PolyA-seq evidence
219 (Fig 2B, S5B Fig). Although Terminitor did not predict the highest number of high confidence
220 poly(A) CS overall, it is always the pipeline having the highest percentage (35.45% and 36.95% for
221 UHR and HBR sample, respectively) of predictions that have PolyA-seq evidence support. This
222 suggests its high precision in detecting novel poly(A) CS when compared to the other methods.
223 Although the raw KLEAT predictions identified a large number of poly(A) CS, only a small
224 proportion can be verified by PolyA-seq. We observed that to maintain precision, certain filters

225 have to be applied to make use of KLEAT predictions, while the choice of filter may be highly
226 library-dependent.

227 We compared the runtime and peak memory usage of all competing tools under the same
228 computing environment (Table 1). All five pipelines used RNA-Bloom [34] as the *de novo* RNA-
229 seq assembler to start with, and the peak memory usage of all pipelines is the same (37.87 GB),
230 contributed by RNA-Bloom. Terminitor, DeepPASTA, DeepGSR, and DeeReCT-PolyA shared the
231 same pipeline to identify candidate sequences, and the only difference is the NN used for
232 sequence classification, which resulted in runtime differences. With a runtime of 7:06:52 hours,
233 DeeReCT-PolyA is the fastest tool, followed by Terminitor with less than 5-minute (1.17%) run
234 time difference. DeepPASTA took two hours more than the first two competitors because of the
235 RNA structure prediction step, although this extra step did not help improve its prediction
236 performance. The long runtime of DeepGSR is probably due to its implementation, which was
237 only compatible with Python 2.7 and older versions of Keras and Theano. KLEAT is more than
238 three times slower compared to all other tools because it involves computationally expensive
239 read-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) CS identification and classification.

Pipeline	Wallclock Runtime (hh:mm:ss)	Wallclock Runtime without assembly
DeeReCT-PolyA	7:06:52	17:29
Terminitor	7:11:23	22:00
DeepPASTA	9:17:49	2:28:26
DeepGSR	11:10:22	4:20:59
KLEAT	25:17:00	18:27:37

240

241 **Impact of expression level on prediction accuracy**

242 The Ensembl annotation is a repertoire of transcript annotations from multiple tissues and cell
243 lines in a range of biological states, so not all of the annotated poly(A) CS are expressed in the
244 HBR and UHR samples, as noted above. We quantified the expression levels of all polyadenylated
245 reference transcripts in Ensembl, and then examined how the four tools perform with respect to
246 poly(A) CS at different expression levels (Fig 3, S6 Fig). Aside from the spikes corresponding to
247 KLEAT raw predictions, Terminitor consistently performs better than all competitors in four tiers
248 of expression thresholds. As expected, the sensitivity of all tools increases as the expression level
249 threshold increases, because unexpressed transcripts are removed from the denominator. The
250 performance difference between Terminitor and other tools increases as well when the
251 expression cut-off increases. In fact, the performance metrics of Terminitor mouse model on
252 poly(A) CS expressed at > 20 TPM are similar to, or better than, DeepPASTA, DeepGSR and
253 DeeReCT-PolyA models for both UHR and HBR samples, showing the promising application of
254 Terminitor on species without a good reference transcriptome annotation.

255 **Fig 3. Performance comparison on the UHR sample with different expression level cut-offs.**
256 Sensitivity and precision of poly(A) CS predictions of Terminitor, KLEAT, DeepPASTA, DeeReCT-
257 PolyA and DeepGSR on expressed Ensembl transcripts. The four facet plots represent the
258 comparison between all expressed transcripts, expressed transcript with TPM > 1, > 10, and > 20,
259 as indicated.

260 **Single nucleotide variation in PAS affects APA choices**

261 One of the advantages of Terminitor is that it is capable of predicting poly(A) CS solely based
262 on sequence context. Consequently, poly(A) CS created or destroyed by base changes can also
263 be identified. We carefully examined all candidate sequences belonging to the UHR sample, and
264 identified sequences containing different PAS hexamer motifs from the reference human
265 genome assembly. Through comparisons between the predicted probabilities and the base

266 variation, we discovered two cases of variation-associated polyadenylation.

267 The first variant base is recorded in dbSNP as rs6484833, a C-to-T change at chr11:36273982
268 (GRCh38.p12) on the reverse strand as seen in Integrative Genomics Viewer (S7A Fig) [35,36].
269 This polymorphism creates the strongest hexamer AATAAA on gene *COMMD9*, and it is
270 associated with an unannotated poly(A) CS at chr11:36273960 on the terminus exon. We
271 speculate that this association may be causal, the former causing the latter. In our pipeline, RNA-
272 Bloom successfully reconstructed this novel transcript to full length with a poly(A) tail, and
273 Terminitor predicted it as a poly(A) CS with 99% probability. This poly(A) CS is also detected in
274 PolyA-Seq of the same sample. Based on Ensembl annotation, *COMMD9* has three poly(A) CS
275 associated with the same terminus exon. This site results in the second-longest 3' UTR, which
276 harbours three miRNA binding sites and extensive RNA binding protein (RBP) sites for 19 RBPs
277 (S8 Fig).

278 The second variant base is rs15342, a T-to-C change at chr15: 101070089 on gene *LRRK1* (S7B
279 Fig). This polymorphism destroys the strongest hexamer AATAAA, which is associated with an
280 annotated poly(A) CS 22 nt downstream. There are 21 sequencing reads containing this
281 polymorphism extend beyond this poly(A) CS, while reads carrying the wildtype are all
282 polyadenylated. RNA-Bloom successfully assembled both the wildtype transcript with a poly(A)
283 tail and the variant transcript without poly(A) tail. Terminitor classifed the wildtype and variant
284 transcripts to contain poly(A) CS with a probability of 98% and 1%, respectively. This
285 polymorphism may prevent cleavage at the poly(A) CS immediately downstream, leading to the
286 alternative usage of the distal poly(A) CS, and producing a transcript with longer 3' UTR. Similarly,
287 extensive (16) miRNA binding sites and RBP sites for 77 RPBs are also predicted to exist on this

288 extended 3' UTR (S9 Fig).

289 **Terminitor performance on long reads**

290 Although the current short read sequencing followed by *de novo* transcriptome assembly is
291 part of Terminitor's mainstream RNA-seq analysis pipeline, the advent of long read sequencing
292 technologies provides unprecedented opportunity to study APA at the transcript isoform level.
293 Here we applied the Terminitor pipeline on two sequence libraries from the human sample
294 NA12878, namely Illumina short reads and PacBio circular consensus sequence (CCS) reads (see
295 Methods for details). Terminitor consistently profiles three times or more poly(A) CS in the PacBio
296 library as in the Illumina library, even under different expression level cut-offs (Fig 4). However,
297 with the Illumina library, Terminitor is able to detect more poly(A) CS at a higher precision (>
298 70%). As the probability cut-off becomes more stringent, less positive poly(A) CS are predicted
299 by both libraries, but the overlapping percentage with respect to the Illumina library increases
300 (S10 Fig). For predicted poly(A) CS with a probability > 90%, 59.27% of the CSs found in the
301 Illumina library are also found in the PacBio library.

302 **Fig 4. Performance comparison between PacBio library and Illumina library on a *H. sapiens***
303 **NA12878 sample.** Sensitivity and precision of poly(A) CS predicted by Terminitor on PacBio and
304 Illumina libraries at different expression level cutoffs. The four facet plots represent the
305 comparison between all expressed transcripts, expressed transcript TPM > 1, > 10, and > 20, as
306 indicated.

307 Discussion and Conclusions

308 Here we introduce a novel deep learning model named Terminitor to detect and classify 3' UTR
309 cleavage sites with or without poly(A) tails, and a profiling pipeline that starts from raw RNA-seq
310 reads to characterize expressed poly(A) CS. The novelties of our model and contributions in this
311 work are as follows:

312 1. The use of concatenated *k*-mer representations instead of raw sequence as model input.
313 Previous applications of deep learning in omics research have shown that designing a proper
314 representation of the raw data is critical to facilitate classification performance. Inputting
315 additional information, namely well-studied sequence features, can also alleviate the burden of
316 learning complex structures. In the case of polyadenylation, several *cis*-acting elements have
317 been revealed, such as the GU-rich sequences in the downstream, UGUA elements in the
318 upstream, and the conserved hexamer motif. Inspired by the usage of *k*-mer in the
319 genome/transcriptome assembly tasks, Terminitor uses *k*-mer representations of raw sequences
320 to mimic functional motifs. Since the auxiliary elements may have different lengths, multiple *k*-
321 mer representations are jointly used to, and are observed to push the accuracy higher.
322 2. The use of an embedding layer to learn the similarities between *k*-mer pairs. One-hot
323 encoded *k*-mer representation of raw sequence is high dimensional and sparse. In our case, the
324 input dimension is exponential to *k*, and most positions in the input vector are zeros. This is
325 computationally inefficient, and would not capture similarities between *k*-mers. To address this,
326 we borrowed the concept of word embedding from natural language processing and used it as
327 an interface between the input layer and the subsequent hidden layers. Essentially, in the
328 embedding layer, one-hot encoded *k*-mers are converted into dense vectors, representing the
329 projection of *k*-mers in a continuous vector space. Consequently, motifs with similar functions
330 are likely to have similar weights, and treated similarly in the subsequent layers.
331 3. Separation of non-polyadenylated CSs from polyadenylated CSs. To our knowledge,
332 Terminitor is the first poly(A) CS classifier that includes non-polyadenylated CS label. Although
333 most protein-coding and long-noncoding RNAs contain poly(A) tails at their 3' ends, a significant

334 number of functional transcripts are not polyadenylated [37]. Since the 3' end processing
335 mechanism of this group is distinct from polyadenylated CS group, separating it out as a new
336 class improves the performance of the model. Sequence clustering based on Terminitor's last
337 hidden layer also groups sequences containing non-polyadenylated CS and non-CS together as a
338 big cluster (S3 Fig).

339 Compared to targeted 3' end sequencing technologies, RNA-seq data, whether generated
340 using short or long read techniques, are more accessible and widely used in transcriptome studies.
341 In the reported work, we demonstrate that the Terminitor pipeline can identify poly(A) CS on
342 expressed transcripts with high precision in comparison to the state-of-the-art methods for RNA-
343 seq data. Failure in detecting a real and expressed poly(A) CS is mostly due to the failure of
344 reconstructing transcripts to their 3' ends. For instance, in the UHR sample, among the 58,925
345 expressed poly(A) CS that were not detected, more than half (30,181) have a TPM less than one,
346 and the majority of them have very few or no reads supporting the poly(A) tail because of
347 sequencing bias.

348 We observed that false positive predictions by Terminitor generally occur for three reasons.
349 The first and most common scenario is mis-assemblies due to the low sequence complexity of
350 the 3' UTR. Second, these poly(A) CS are in fact real novel events, but unannotated; around 30%
351 of the Terminitor false positives (using the Ensembl annotation) are corroborated by PolyA-Seq
352 evidence (Fig 2B, S5B Fig). Third, mis-alignment of upstream sequences, meaning the sequence
353 may contain a real poly(A) CS, but the genomic location is incorrect. In the Terminitor pipeline,
354 when an upstream sequence is multi-mapped, only the primary alignment is chosen to help
355 reduce the false positive rate. However, when the sequence comes from gene paralogs, the

356 genomic location inferred from the chosen alignment does not necessarily correspond to that of
357 the expressed paralog.

358 In principle, the Terminitor pipeline can be applied on any reconstructed transcripts to
359 determine if its 3'-end contains a poly(A) CS, or non-polyadenylated CS, or no site (3' incomplete).
360 Aside from benchmarking its performance on traditional Illumina short read assemblies, we also
361 explored its application on PacBio long read technology. Terminitor demonstrated more than
362 twice the CS prediction sensitivity when using the PacBio vs. Illumina library, probably because
363 more sequences containing real poly(A) CS are fed into the NN model. PacBio CCS reads are self-
364 corrected through consensus calling step, resulting in near- or full-length transcripts with poly(A)
365 tails. In contrast, transcripts reconstructed from an Illumina library may suffer from mis-
366 assemblies, and lowly expressed poly(A) CS may be obscured by longer transcripts. In our analysis,
367 among the 548,448 candidate sequences extracted from the PacBio library, 29.71% of them
368 contain untemplated poly(A)s, while the ratio is much lower (7.22%) in 381,934 assembled
369 sequences using the Illumina library.

370 In summary, Terminitor outperforms all state-of-the-art methods in the datasets we
371 benchmarked them on, and executes within a reasonable time, making it an attractive tool to
372 complement other RNA-seq analysis pipelines. The performance of Terminitor is robust on both
373 short and long-read sequencing technologies. As long-read RNA-seq finds broader uptake in
374 transcriptome studies, it is poised to shed light on the association between poly(A) CS and
375 transcript isoforms, fueling a new era of APA research. Terminitor is a candidate enabling
376 technology for such studies. Further, we have observed how Terminitor can identify novel poly(A)
377 CS caused by base variations. As has been elucidated in previous studies, novel poly(A) CS are

378 expressed in cancer tissues, and with the help of Terminitor, mechanisms and impacts of APA can
379 be better understood. Even for species without an established poly(A) CS database, their
380 expressed poly(A) CS can be identified with Terminitor using transfer learning, where the model
381 is trained on a different but related species. The usage of Terminitor, as a sequence classifier, is
382 not limited to poly(A) CS profiling, but can also be extended to facilitate gene annotation and
383 testing the completeness of assembled transcripts. We expect Terminitor to have broad
384 applications in genome/transcriptome annotation, novel isoform identification, APA analysis and
385 gene regulation studies.

386 ## Methods

387 ### Datasets

388 To train and validate the NN model, and to demonstrate transfer learning, we collated two
389 reference datasets describing human and mouse transcriptomes, based on poly(A) CS databases
390 and Ensembl annotations. Each dataset is composed of sequences with equal lengths coming
391 from three classes: poly(A) CS, non-poly(A) CS, and non-CS (Table 2). PolyA_DB version3.1 was
392 used as our primary poly(A) CS library [13]. We first mapped poly(A) CS recorded in PolyA_DB3
393 to Ensembl annotation release 94 using `bedtools closest` (v2.27.1) and selected the most
394 compatible transcript for each site. Poly(A) CS supported by less than 1 Reads Per Million (RPM)
395 or located more than 1000 nt downstream of any annotated transcript were considered
396 unreliable and were discarded.

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

Species	poly(A) CS	non-poly(A) CS	non-CS
H. sapiens	52,457	27,595	52,460
M. musculus	75,661	37,028	75,660

397

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

404 Non-CS comprises five types of genomic locations: exonic, intronic, intergenic, 5' and 3'
405 untranslated regions, and are randomly selected at least 100 nt away from any annotated 3' ends,
406 including Ensembl annotation and compiled poly(A) library.

407 Our datasets are imbalanced in their label abundances due to the limited number of non-
408 polyadenylated CS. However, to fully exploit the potential of deep NNs, we used all non-
409 redundant poly(A) CS and CS generated as above descriptions, and deliberately matched the
410 number of pseudo sites to the number of poly(A) CS. Because the label abundances are
411 imbalanced, we used both accuracy and the F-measure, the harmonic mean of precision and
412 recall, as assessment metrics during model training and benchmarking.

413 **Model training**

414 The NN model is implemented in Python 3.6.6, using Keras library 2.2.4 [38] with TensorFlow
415 1.11.0 [39] as the backend. To evaluate the performance of the model with different
416 hyperparameter combinations and input sequence lengths, we used stratified 5-fold cross
417 validation on the whole dataset to monitor the overall accuracy of three labels, and accuracy,
418 sensitivity, and specificity on poly(A) label. All performance metrics on the training sets reported

419 in this study are the mean and standard deviation of five validation runs. In each run, 80% of data
420 is used for training and 20% is used for validation. Next, we trained the model using the entire
421 training set and used it for benchmarking the performance on experimental data.

422 **Model architecture and hyperparameter tuning**

423 Searching for the optimal hyperparameters for a deep NN has always been a tedious task, as
424 many parameters are dependent on each other in the way they influence performance. We first
425 built the basic frame with one input layer, one embedding layer, and one output classification
426 layer, and then fine-tune hyperparameters related to architecture and training using grid search,
427 as listed in Table 3. For all models, initial model weights were randomly drawn from Glorot
428 uniform initializer [40]. The weights of the three neurons in the output layer are computed via
429 softmax activation function, representing the probability distribution of the three labels, and they
430 always add up to one. To avoid over-fitting, we applied early stopping in training each model. As
431 an iterative method, the validation loss is monitored at each training epoch, and when it stays
432 the same or increases for the next ten epochs, the training process is stopped. Without early
433 stopping, too many training epochs will lead to nearly 100% accuracy on training set, yet yield
434 very low accuracy on validation and test sets.

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 layers	4, 16, **32**, **64**, **128**, 256, **512**, 1024 (Refer to Fig 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 Terminitor is depicted in Fig 5 and the selected optimal hyperparameters are in bold font in Table 3. The first step of Terminitor 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, where

$$l = \sum_{k \in \{4,6,8,10\}} 4^k \qquad (1)$$

Fig 5. Architecture of Terminitor. The information flows from bottom up. During training stage, reference transcripts are aligned back to the reference genome, and flanking sequence on both side of the poly(A) CS is extracted (grey thick sequence). The sequence is *k*-merized and four *k* values are used. Then all *k*-mer representations are concatenated and each *k*-mer is one-hot encoded (each square represents one *k*-mer). The input sequences are fed into an embedding layer, followed by three fully connected layers, and then the probability of belonging to one of the three classes is calculated through a Softmax function.

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, and the activation function between each adjacent layer is the rectified linear unit (ReLU) function. Since dropout regularization did not achieve better performance, it was not performed in the final model.

Experimental data

452 The raw RNA-seq reads for the two MicroArray Quality Control project reference human samples,
453 HBR and UHR, were downloaded from Illumina BaseSpace, and each sample has two replicates
454 (https://basespace.illumina.com/projects/3777775/samples) [41]. RNA-seq libraries of the four
455 samples were prepared with Illumina TruSeq Stranded kits using poly(A) selection, and
456 sequenced as 75bp paired-end reads. The Illumina and PacBio reads for NA12878 were obtained
457 from an earlier study under accession number SRR1153470 and SRR1163655, respectively [42].
458 The Illumina reads are from a poly(A) selected library sequenced with 75 bp stranded paired-end
459 reads. The PacBio library was prepared from the same batch of poly(A) selected RNA and
460 sequenced with PacBio CCS method. The filtered PolyA-Seq sites were downloaded from the NCBI
461 Gene Expression Omnibus (GEO) under study GSE30198. We combined sites detected by the two
462 HBR replicates (GSM747473 and GSM GSM747474), and sites detected by the two UHR replicates
463 (GSM747475 and GSM747476) together respectively for benchmarking analysis.

464 **Prediction pipeline**

465 We also provide a poly(A) CS prediction pipeline based on Terminitor. When using Illumina short
466 reads, the pipeline starts by assembling RNA-seq data using RNA-Bloom [34]. RNA-Boom is run
467 with the option `-stratum 01`, which allows extension of all fragments regardless of its
468 coverage, and the option `--polya`, which prioritizes the assembly of transcripts with poly(A)
469 tails. The false positive rate (FPR) of Bloom filters in the program is set to 0.005. Then, all
470 assembled transcripts are aligned to the reference genome hg38 and compared to Ensembl
471 annotation release 94 to select sequences with untemplated As at their 3' ends, or sequences
472 that end in the 3' UTR of annotated transcripts. Similar to the structure of sequences in the
473 training dataset, selected sequences are used to construct test sequences, each composed of

474 three parts: a fixed length (100 nt) of upstream sequence extracted from the assembled
475 transcript, a potential cleavage site, and a fixed length (100 nt) of downstream genomic sequence
476 extracted from the reference. Finally, these test sequences are fed into the trained Terminitor
477 model for classification.

478 **Evaluation of prediction pipeline**

479 Poly(A) CS are often observed as a group of neighboring sites rather than a precise genomic
480 location [7]. To accommodate, predicted poly(A) CS within 30 nt of each other are clustered, and
481 the one with the highest probability score is chosen. Ensembl annotated poly(A) CS are also
482 clustered in the same way, and the one located in the median position is chosen as the
483 representative. Then, the predicted list is compared with the annotated list to compute hits and
484 misses. In our evaluation system, a true positive poly(A) CS is a predicted one that lies within ±
485 30 nt of an annotated site; a false positive site is a predicted poly(A) CS that does not lie within ±
486 30 nt of any annotated ones; false negative means an annotated site does not have any predicted
487 site within ± 30 nt. For this analysis, true negative (no real or predicted poly(A) CS in a certain
488 location) is not informative for experimental data. Using these definitions, we report sensitivity
489 and precision as performance measures for all experimental data tests. Note that in the training
490 data, true negative is informative, and we also report accuracy for our cross validation results.

491 **Tool comparison**

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

495 KLEAT pipeline was chosen as a representative of read-evidence based approaches. The
496 auxiliary files KLEAT uses include: 1) RNA-seq assemblies from RNA-Bloom, 2) read to contig
497 alignment files generated by BWA-MEM 0.7.17-r1188 [43], and 3) contig to genome alignment
498 files generated by GMAP version 2017-01-14 [44]. Since KLEAT only works for the human genome
499 reference hg19, all predicted poly(A) CS were converted into hg38 genomic coordinates by UCSC
500 liftOver [45]. KLEAT outputs all possible poly(A) CS and their supporting evidence, such as the
501 length of poly(A) tail in reconstructed transcripts, the number of reads containing poly(A) tail,
502 and the length of poly(A) tail of reads that can be mapped to a reconstructed transcript. We
503 applied filters to the raw predictions with different combinations of three evidence types and
504 computed the Pareto frontier on the sensitivity-precision plane.

505 Two recent deep learning models, DeepGSR and DeeReCT-PolyA, were also included in the
506 comparison. Like Terminitor, these two tools are NN models expecting fixed length sequences as
507 input. Starting from all candidate sequences processed from our prediction pipeline, we selected
508 only the ones containing PAS as these tools expects PAS in the input sequences. Then we adjusted
509 the sequence lengths as required by the two models (± 300 nt around the PAS for DeepGSR; ±
510 100 nt around the PAS for DeeReCT-PolyA), and fed into these two models for classification. For
511 DeepGSR, we used two pre-trained models from its publication, one only on the AATAAA
512 hexamer, and the other on all 16 hexamers [29]. Similarly, the DeeReCT-PolyA package provides
513 two pre-trained models based on two human datasets, dragon and omni, both of which were
514 included in the comparison. Both datasets were generated by previous poly(A) prediction
515 researches [46,47]. Both DeepGSR and DeeReCT-PolyA only output a binary value instead of a

516 confidence score that can be tuned with different cut-offs, so their predictions contributed to
517 four data points in the comparison plot: two models for two tools.

518 **Data access**

519 Terminitor is implemented using Python 3.6 with Keras package. The software package is freely
520 available under GNU General Public Licence. The source code and manual are available at our
521 GitHub repository https://github.com/bcgsc/Terminitor. Previously published data used in this
522 work are described in the Methods section, and are available from several public repositories.
523 The human and mouse reference datasets colated in this study and trained models are deposited
524 in public ftp site http://www.bcgsc.ca/downloads/supplementary/Terminitor.

525 ## Reference

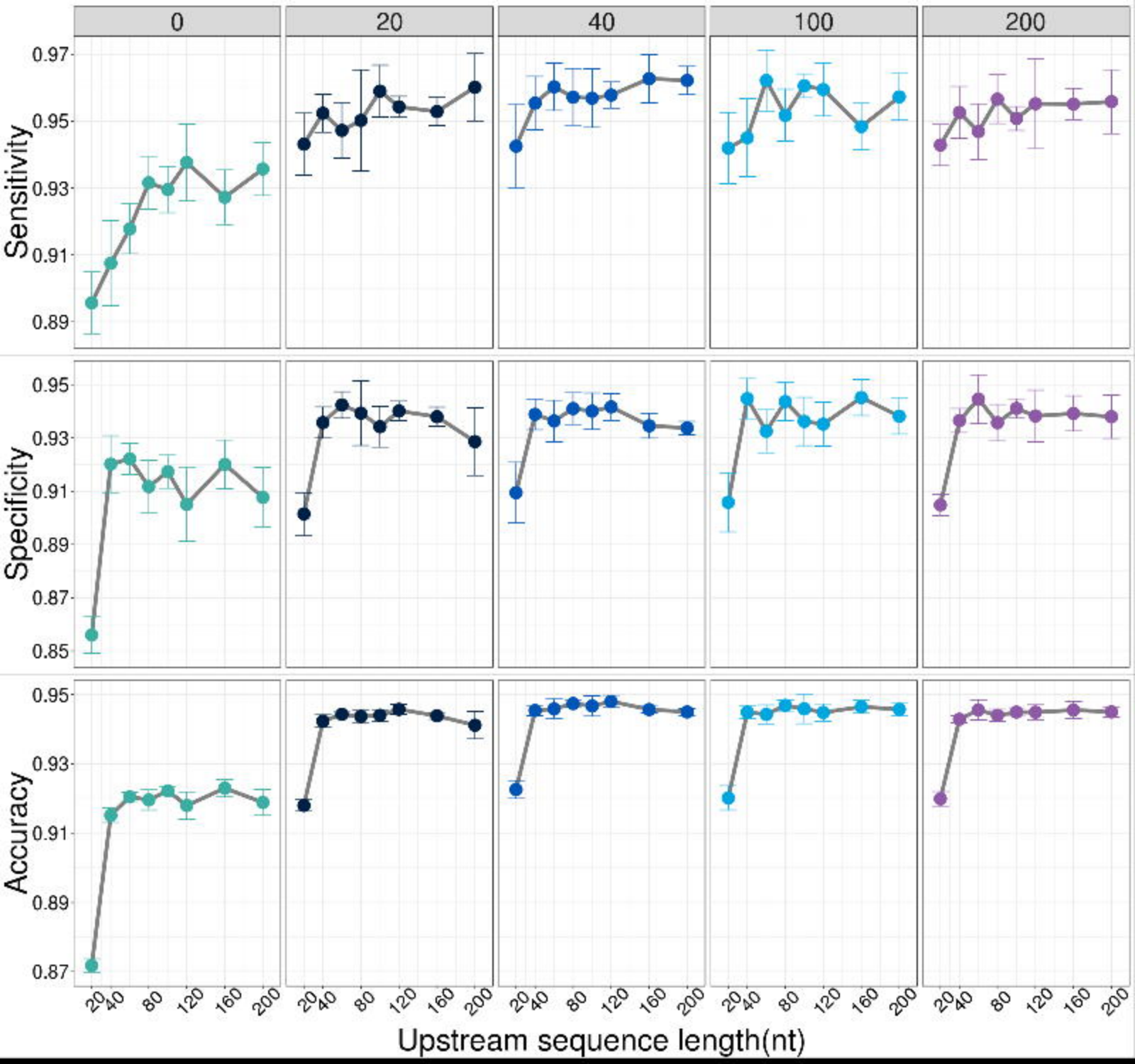
526 1. Edmonds M, Vaughan MH, Nakazato H. Polyadenylic acid sequences in the heterogeneous
527 nuclear RNA and rapidly-labeled polyribosomal RNA of HeLa cells: possible evidence for a
528 precursor relationship. Proc Natl Acad Sci U S A. 1971;68: 1336–40. doi:10.1073/pnas.68.6.1336
529 2. Neve J, Patel R, Wang Z, Louey A, Furger AF. Cleavage and polyadenylation: Ending the message
530 expands gene regulation. RNA Biol. 2017;14: 865–890.
531 3. Barrett LW, Fletcher S, Wilton SD. Regulation of eukaryotic gene expression by the untranslated
532 gene regions and other non-coding elements. Cellular and Molecular Life Sciences. 2012. pp.
533 3613–3634. doi:10.1007/s00018-012-0990-9
534 4. Shi Y, Chan S, Martinez-Santibañez G. An up-close look at the pre-mRNA 3'-end processing
535 complex. RNA Biol. 2009;6: 522–525. doi:10.4161/rna.6.5.9554
536 5. Zhang H, Lee JY, Tian B. Biased alternative polyadenylation in human tissues. Genome Biol.
537 2005;6: R100. doi:10.1186/gb-2005-6-12-r100
538 6. Tian B, Graber JH. Signals for pre-mRNA cleavage and polyadenylation. Wiley Interdisciplinary
539 Reviews: RNA. 2012. pp. 385–396. doi:10.1002/wrna.116
540 7. Tian B, Hu J, Zhang H, Lutz CS. A large-scale analysis of mRNA polyadenylation of human and
541 mouse genes. Nucleic Acids Res. 2005;33: 201–212. doi:10.1093/nar/gki158
542 8. Lianoglou S, Garg V, Yang JL, Leslie CS, Mayr C. Ubiquitously transcribed genes use alternative
543 polyadenylation to achieve tissue-specific expression. Genes Dev. 2013;27: 2380–2396.
544 doi:10.1101/gad.229328.113
545 9. Lembo A, Di Cunto F, Provero P. Shortening of 3'UTRs correlates with poor prognosis in breast
546 and lung cancer. PLoS One. 2012;7. doi:10.1371/journal.pone.0031129
547 10. Li L, Wang D, Xue M, Mi X, Liang Y, Wang P. 3'UTR shortening identifies high-risk cancers with
548 targeted dysregulation of the ceRNA network. Sci Rep. 2015;4: 5406. doi:10.1038/srep05406
549 11. Xue Z, Warren RL, Gibb EA, MacMillan D, Wong J, Chiu R, et al. Recurrent tumor-specific
550 regulation of alternative polyadenylation of cancer-related genes. BMC Genomics. 2018;19.
551 doi:10.1186/s12864-018-4903-7

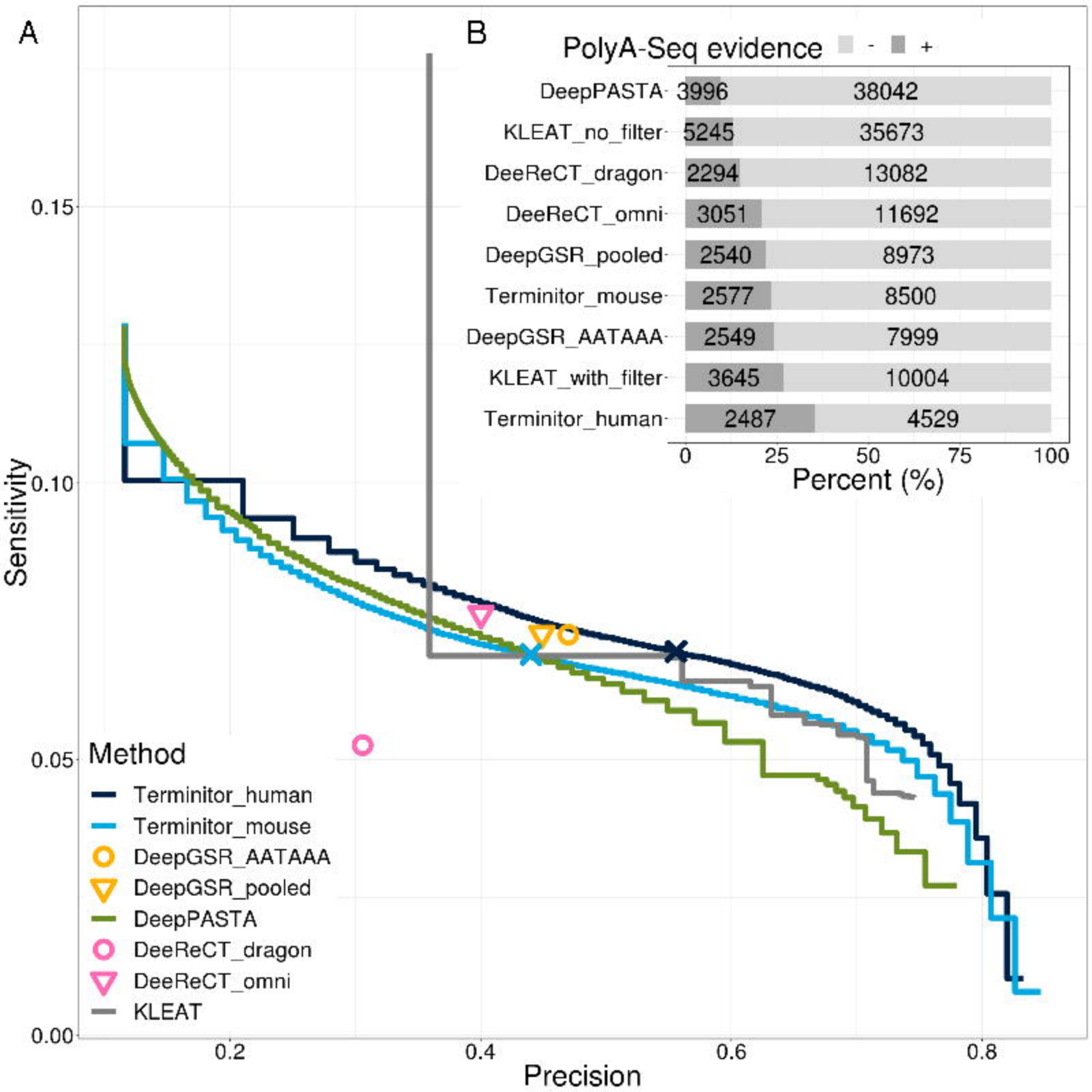
552 12. Aken BL, Ayling S, Barrell D, Clarke L, Curwen V, Fairley S, et al. The Ensembl gene annotation
553 system. Database. 2016;2016: baw093. doi:10.1093/database/baw093
554 13. Wang R, Nambiar R, Zheng D, Tian B. PolyA-DB 3 catalogs cleavage and polyadenylation sites
555 identified by deep sequencing in multiple genomes. Nucleic Acids Res. 2018;46: D315–D319.
556 doi:10.1093/nar/gkx1000
557 14. Liu D, Brockman JM, Dass B, Hutchins LN, Singh P, McCarrey JR, et al. Systematic variation in
558 mRNA 3'-processing signals during mouse spermatogenesis. Nucleic Acids Res. 2007;35: 234–246.
559 doi:10.1093/nar/gkl919
560 15. Mayr C. Evolution and Biological Roles of Alternative 3'UTRs. Trends in Cell Biology. 2016. pp.
561 227–237. doi:10.1016/j.tcb.2015.10.012
562 16. Shepard PJ, Choi E-A, Lu J, Flanagan LA, Hertel KJ, Shi Y. Complex and dynamic landscape of RNA
563 polyadenylation revealed by PAS-Seq. RNA. 2011;17: 761–772. doi:10.1261/rna.2581711
564 17. Harrison PF, Powell DR, Clancy JL, Preiss T, Boag PR, Traven ANA, et al. PAT-seq : a method to
565 study the integration of 3' -UTR dynamics with gene expression in the eukaryotic transcriptome.
566 Rna. 2015;21: 1502–1510. doi:10.1261/rna.048355.114.
567 18. Chang H, Lim J, Ha M, Kim VN. TAIL-seq: Genome-wide determination of poly(A) tail length and 3'
568 end modifications. Mol Cell. 2014;53: 1044–1052. doi:10.1016/j.molcel.2014.02.007
569 19. Derti A, Garrett-Engele P, MacIsaac KD, Stevens RC, Sriram S, Chen R, et al. A quantitative atlas of
570 polyadenylation in five mammals. Genome Res. 2012;22: 1173–1183. doi:10.1101/gr.132563.111
571 20. Müller S, Rycak L, Afonso-Grunz F, Winter P, Zawada AM, Damrath E, et al. APADB: a database for
572 alternative polyadenylation and microRNA regulation events. Database (Oxford). 2014;2014.
573 doi:10.1093/database/bau076
574 21. You L, Wu J, Feng Y, Fu Y, Guo Y, Long L, et al. APASdb: A database describing alternative poly(A)
575 sites and selection of heterogeneous cleavage sites downstream of poly(A) signals. Nucleic Acids
576 Res. 2015;43: D59–D67. doi:10.1093/nar/gku1076
577 22. Gruber AJ, Schmidt R, Gruber AR, Martin G, Ghosh S, Belmadani M, et al. A comprehensive
578 analysis of 3' end sequencing data sets reveals novel polyadenylation signals and the repressive
579 role of heterogeneous ribonucleoprotein C on cleavage and polyadenylation. Genome Res.
580 2016;26: 1145–1159. doi:10.1101/gr.202432.115
581 23. Birol I, Raymond A, Chiu R, Nip KMi, Jackman SD, Kreitzman M, et al. Kleat: Cleavage Site Analysis
582 of Transcriptomes. Biocomput 2015. 2014; 347–358. doi:10.1142/9789814644730_0034
583 24. Bonfert T, Friedel CC. Prediction of poly(A) sites by poly(A) read mapping. PLoS One. 2017;12.
584 doi:10.1371/journal.pone.0170914
585 25. Xia Z, Donehower LA, Cooper TA, Neilson JR, Wheeler DA, Wagner EJ, et al. Dynamic analyses of
586 alternative polyadenylation from RNA-seq reveal a 3'-UTR landscape across seven tumour types.
587 Nat Commun. 2014;5: 5274. doi:10.1038/ncomms6274
588 26. Ye C, Long Y, Ji G, Li QQ, Wu X. APAtrap: Identification and quantification of alternative
589 polyadenylation sites from RNA-seq data. Bioinformatics. 2018;34: 1841–1849.
590 doi:10.1093/bioinformatics/bty029
591 27. Ha KCH, Blencowe BJ, Morris Q. QAPA: A new method for the systematic analysis of alternative
592 polyadenylation from RNA-seq data. Genome Biol. 2018;19. doi:10.1186/s13059-018-1414-4
593 28. Xia Z, Li Y, Zhang B, Li Z, Hu Y, Chen W, et al. DeeReCT-PolyA: A robust and generic deep learning
594 method for PAS identification. Bioinformatics. 2019;35: 2371–2379.
595 doi:10.1093/bioinformatics/bty991
596 29. Kalkatawi M, Magana-Mora A, Jankovic B, Bajic VB. DeepGSR: an optimized deep-learning
597 structure for the recognition of genomic signals and regions. Bioinformatics. 2018.
598 doi:10.1093/bioinformatics/bty752
599 30. Arefeen A, Xiao X, Jiang T. DeepPASTA: deep neural network based polyadenylation site analysis.

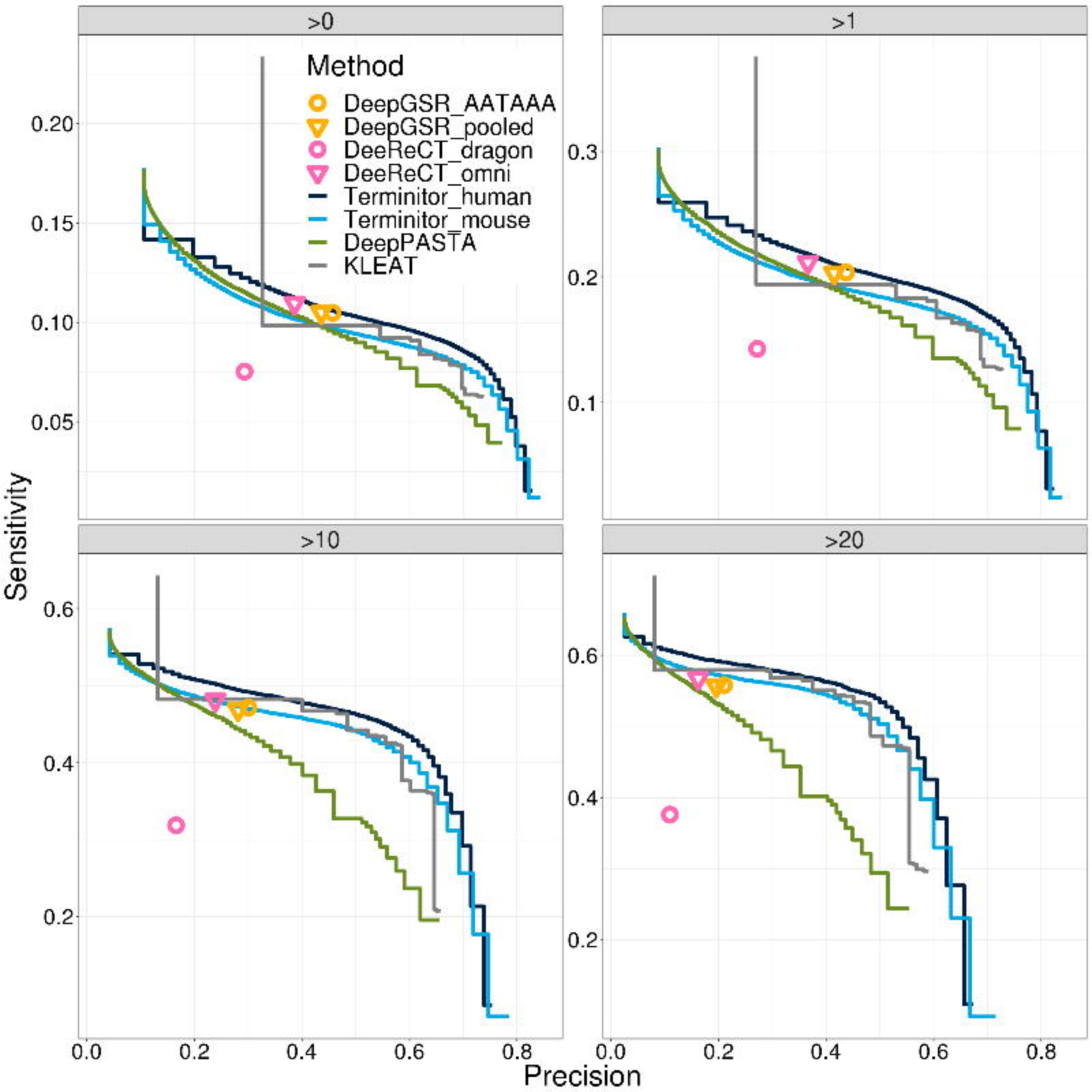
Bioinformatics. 2019. doi:10.1093/bioinformatics/btz283
31. Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in bioinformatics. 2017. pp. 851–869. doi:10.1093/bib/bbw068
32. Venkataraman K, Brown KM, Gilmartin GM. Analysis of a noncanonical poly(A) site reveals a tripartite mechanism for vertebrate poly(A) site recognition. Genes Dev. 2005;19: 1315–1327. doi:10.1101/gad.1298605
33. 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: 1523–1536. doi:10.1038/emboj.2010.42
34. 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. 2019; 701607. doi:10.1101/701607
35. Smigielski EM. dbSNP: a database of single nucleotide polymorphisms. Nucleic Acids Res. 2000;28: 352–355. doi:10.1093/nar/28.1.352
36. Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): High-performance genomics data visualization and exploration. Brief Bioinform. 2013;14: 178–192. doi:10.1093/bib/bbs017
37. Yang L, Duff MO, Graveley BR, Carmichael GG, Chen LL. Genomewide characterization of non-polyadenylated RNAs. Genome Biol. 2011;12. doi:10.1186/gb-2011-12-2-r16
38. Chollet F. Keras: Deep Learning for humans. In: Github [Internet]. 2015. Available: https://github.com/keras-team/keras
39. 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; 272–283.
40. Glorot X, Statistics YBBT-P of the TIC on AI and. Understanding the difficulty of training deep feedforward neural networks. Teh YW, Titterington M, editors. PMLR ; 2010. pp. 249–256. Available: http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf
41. 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: 1151–1161. doi:10.1038/nbt1239
42. 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: 9869–9874. doi:10.1073/pnas.1400447111
43. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv Prepr arXiv. 2013;00: 3. doi:10.1186/s13756-018-0352-y
44. Wu TD, Watanabe CK. GMAP: A genomic mapping and alignment program for mRNA and EST sequences. Bioinformatics. 2005;21: 1859–1875. doi:10.1093/bioinformatics/bti310
45. Hinrichs AS, Karolchik D, Baertsch R, Barber GP, Bejerano G, Clawson H, et al. The UCSC Genome Browser Database: update 2006. Nucleic Acids Res. 2006;34: D590–D598. doi:10.1093/nar/gkj144
46. Kalkatawi M, Rangkuti F, Schramm M, Jankovic BR, Kamau A, Chowdhary R, et al. Dragon polya spotter: Predictor of poly(A) motifs within human genomic DNA sequences. Bioinformatics. 2012;28: 127–129. doi:10.1093/bioinformatics/btr602
47. Magana-Mora A, Kalkatawi M, Bajic VB. Omni-Polya: A method and tool for accurate recognition of poly(A) signals in human genomic DNA. BMC Genomics. 2017;18. doi:10.1186/s12864-017-4033-7

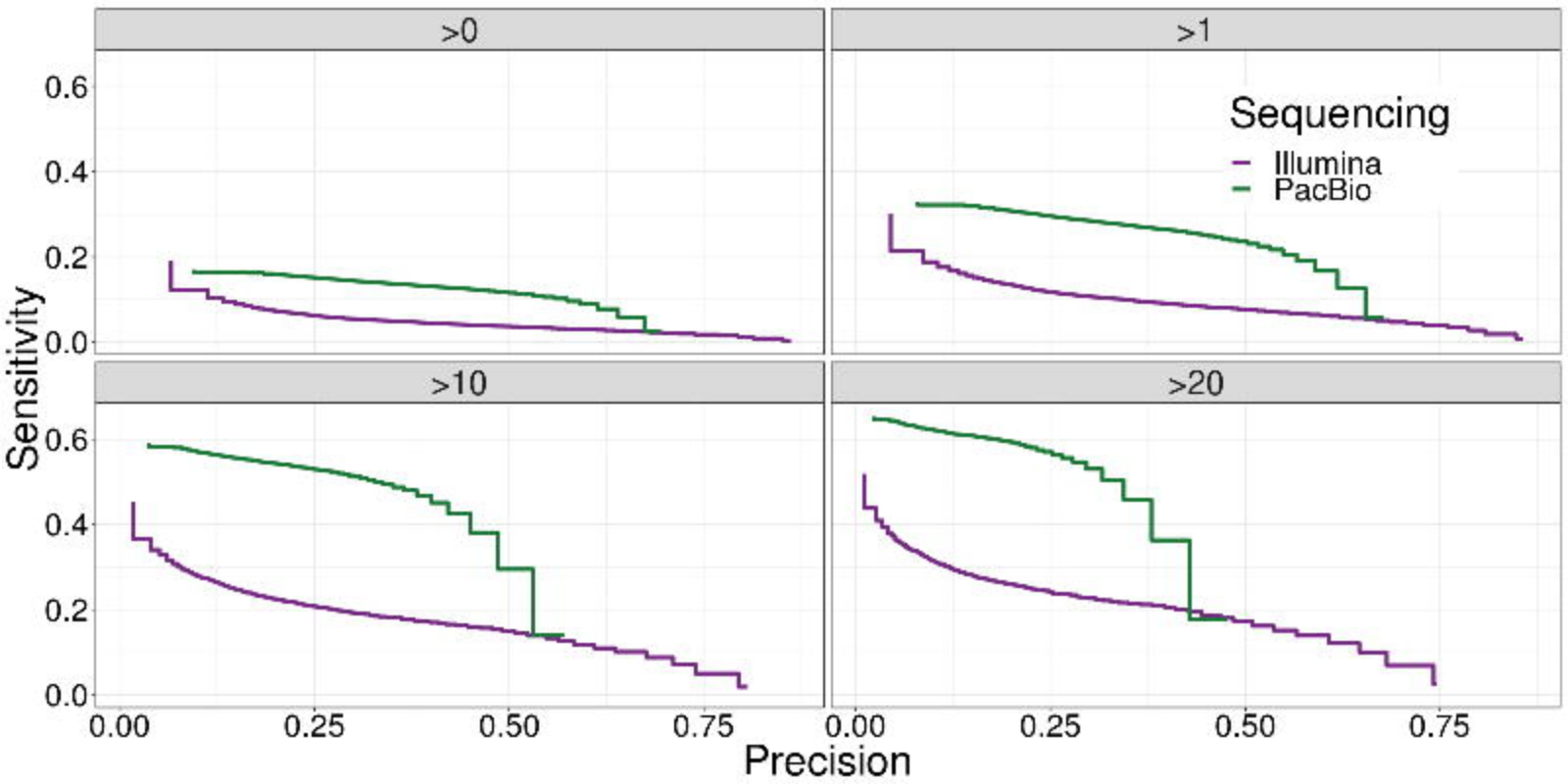
647 ## Supporting information

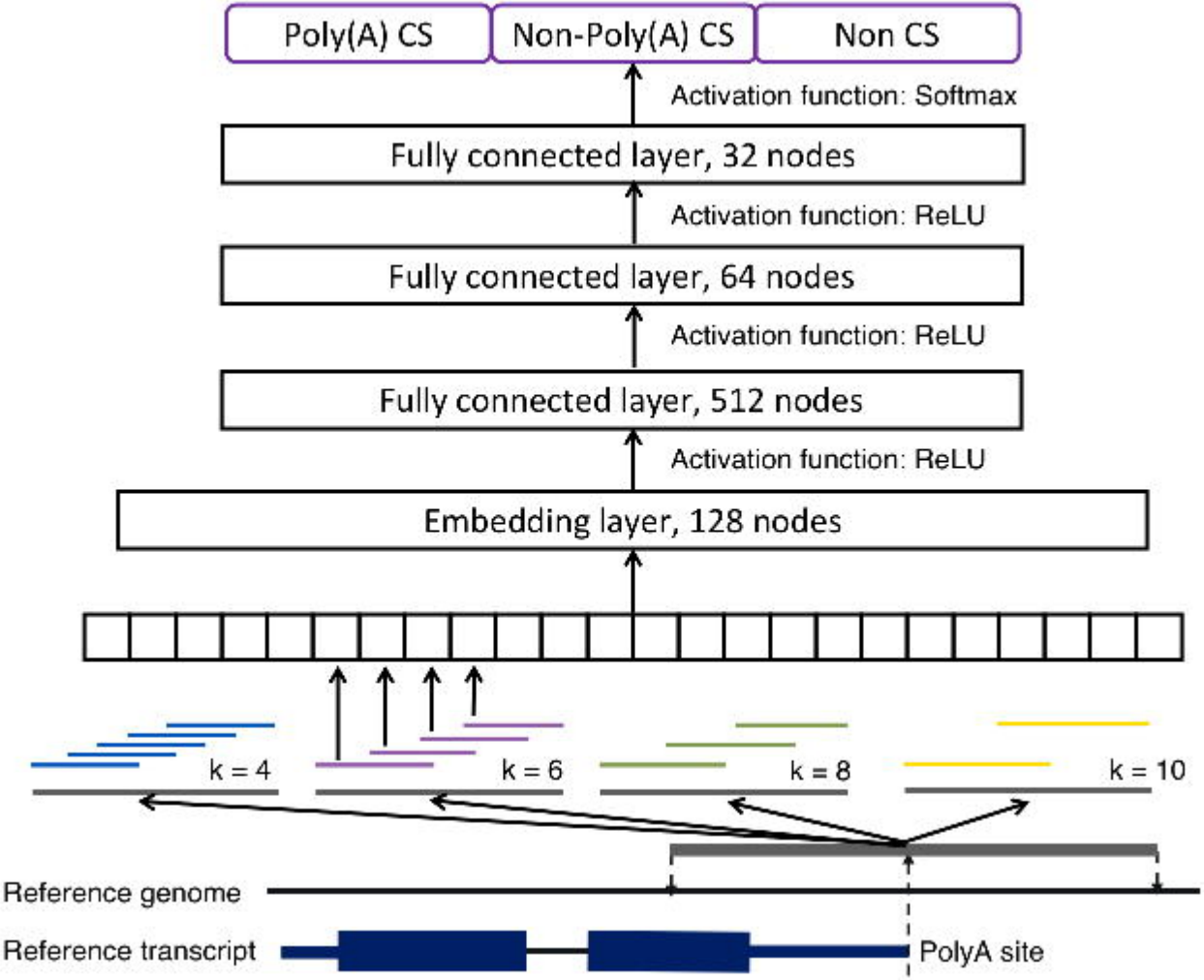
648 **S1 File. Supplementary material.** This file contains all supplementary figures S1-10 Fig.


0
20
40
100
200
Sensitivity
0.97
0.95
0.93
0.91
0.89
Specificity
0.95
0.93
0.91
0.89
0.87
0.85
Accuracy
0.95
0.93
0.91
0.89
0.87
20
40
80
120
160
200
Upstream sequence length(nt)


A
B
PolyA-Seq evidence
-
+
DeepPASTA
3996
38042
KLEAT_no_filter
5245
35673
DeeReCT_dragon
2294
13082
DeeReCT_omni
3051
11692
DeepGSR_pooled
2540
8973
Terminitor_mouse
2577
8500
DeepGSR_AATAAA
2549
7999
KLEAT_with_filter
3645
10004
Terminitor_human
2487
4529
0
25
50
75
100
Percent (%)
Sensitivity
0.15
0.10
0.05
0.00
0.2
0.4
0.6
0.8
Precision
Method
Terminitor_human
Terminitor_mouse
DeepGSR_AATAAA
DeepGSR_pooled
DeepPASTA
DeeReCT_dragon
DeeReCT_omni
KLEAT


>0
>1
>10
>20
Method
DeepGSR_AATAAA
DeepGSR_pooled
DeeReCT_dragon
DeeReCT_omni
Terminitor_human
Terminitor_mouse
DeepPASTA
KLEAT
Sensitivity
Precision
0.05
0.10
0.15
0.20
0.1
0.2
0.3
0.4
0.6
0.0
0.2
0.4
0.6
0.8


>0
>1
>10
>20
Sequencing
Illumina
PacBio
Sensitivity
Precision
0.0
0.2
0.4
0.6
0.00
0.25
0.50
0.75


Poly(A) CS
Non-Poly(A) CS
Non CS
Activation function: Softmax
Fully connected layer, 32 nodes
Activation function: ReLU
Fully connected layer, 64 nodes
Activation function: ReLU
Fully connected layer, 512 nodes
Activation function: ReLU
Embedding layer, 128 nodes
k = 4
k = 6
k = 8
k = 10
Reference genome
Reference transcript
PolyA site