Cell Systems

Supplemental Information

Prober: A general toolkit for analyzing sequencing-based toeprinting assays

Bo Li, Akshay Tambe, Sharon Aviran and Lior Pachter

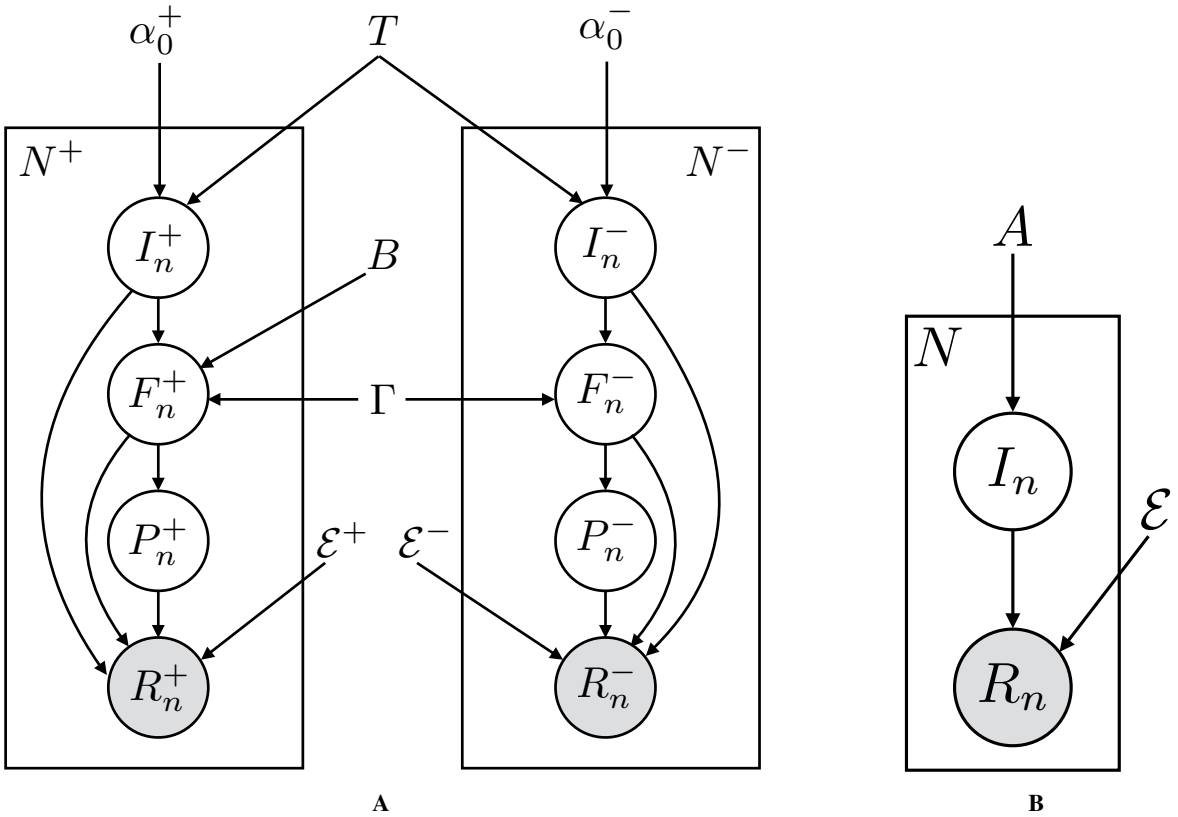

Figure S1. Related to Experimental Procedures. Graphical representations of PROBer's probabilistic generative models. (A) Graphical model for toeprinting assays other than iCLIP. The plate in the left generates treatment data and the right plate generates control data. In each plate, unshaded circle represents hidden variable, shaded circle represents observed variable, and arrow represents dependency. In addition, the letter at the top corner denotes the number of observed reads each group needs to generate. Letters outside the plates are the model parameters we need to learn. To generate a read from the treatment group, we first sample I^+, the isoform that read comes from. Then based on I^+, Γ, and B, we generate F^+, the reverse transcribed framgent that contains the read. If the length of F^+ is in an appropriate range, this fragment passes the size selection and $P^+ = 1$. Otherwise, F^+ fails and $P^+ = 0$. Lastly, if $P^+ = 1$, we generate the read sequence, R^+, according to the sequencing error model, $\mathscr{E}^+$. (B) Graphical model for iCLIP. Unshaded circle represents hidden variable, shaded circle represents observed variable, and arrow represents dependency. To generate an iCLIP read, we first determine the crosslink site I. Then based on I and read generating parameters $\mathscr{E}$, we produce the observed read R.

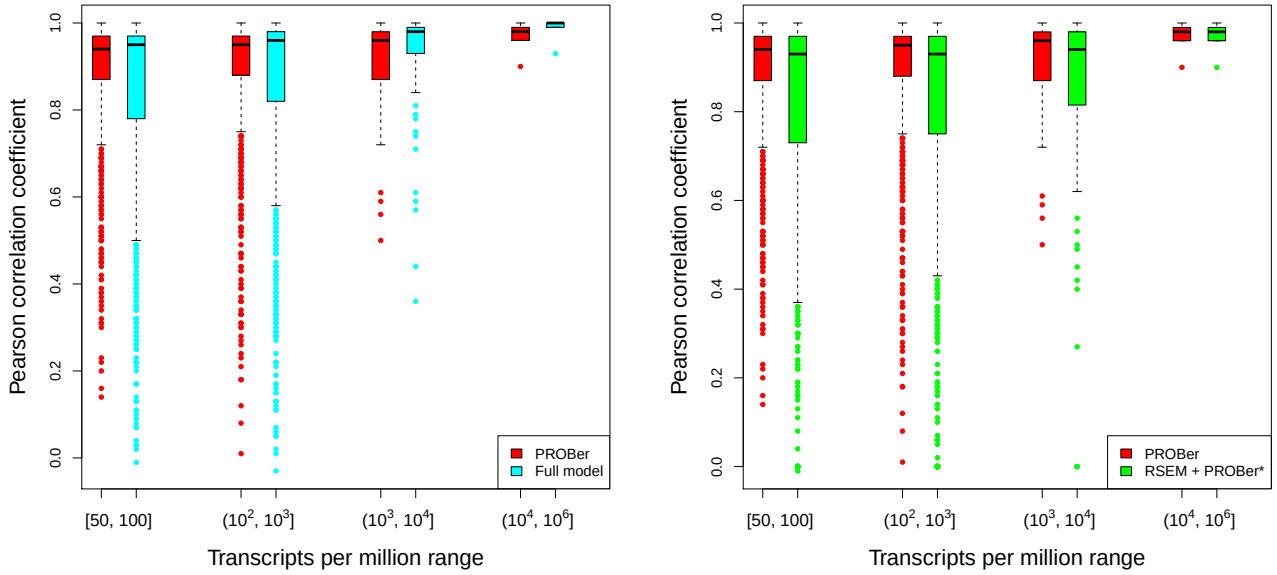

Figure S2. Related to Experimental Procedures. Box plots of Pearson's correlation coefficients comparing `PROBer` with (A) the full model and (B) `RSEM + PROBer*` pipeline on the simulated arabidopsis data set. The number of transcripts in each expression range is: 887 in [50, 100], 849 in (100, 1000], 60 in (1000, 10000], and 6 in (10000, 1000000].

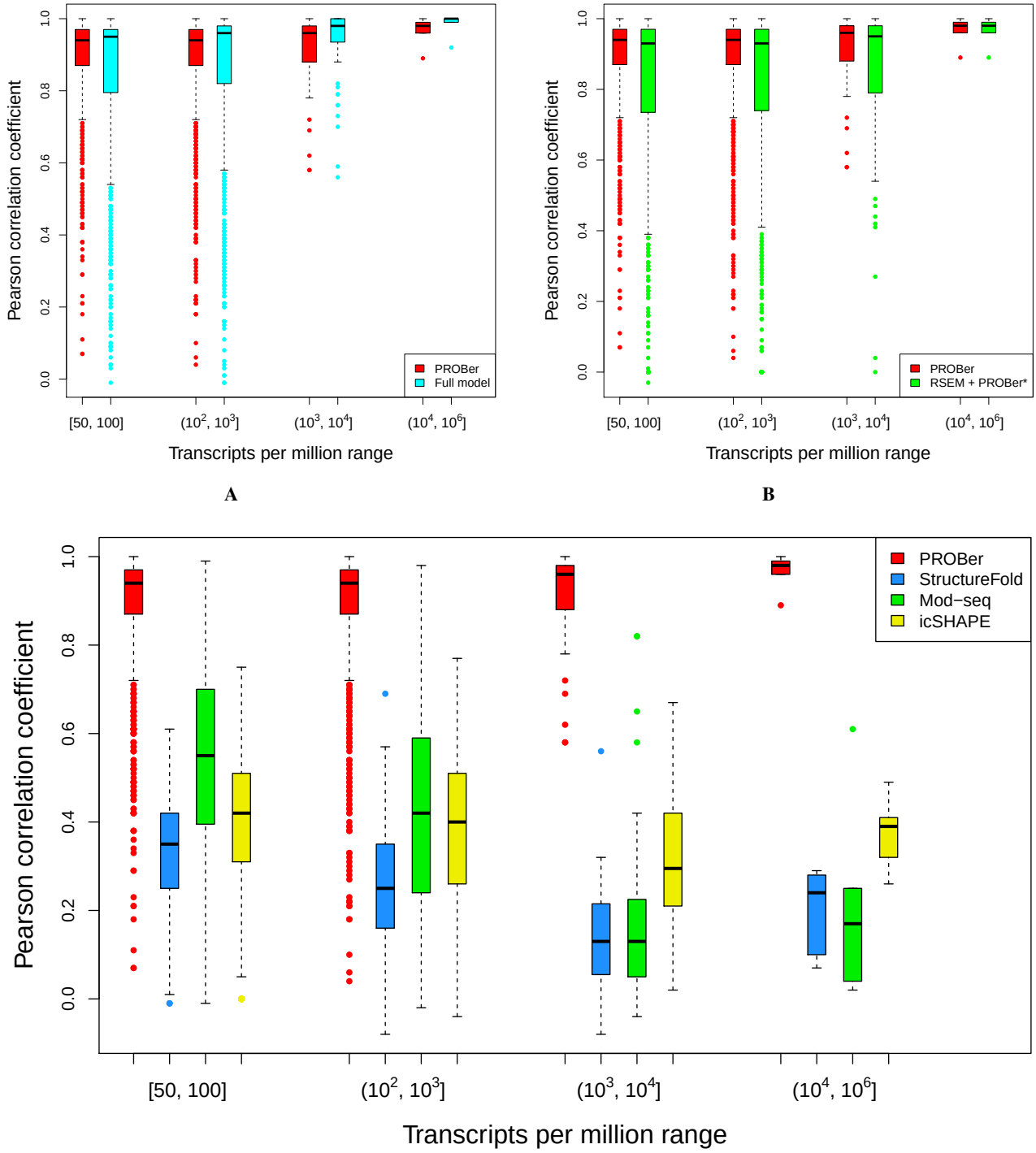

Figure S3. Related to Figure 2A and Experimental Procedures. Box plots of Pearson's correlation coefficients on an extra simulated arabidopsis data set. The number of transcripts in each expression range is: 887 in [50, 100], 849 in (100, 1000], 60 in (1000, 10000], and 6 in (10000, 1000000]. (A) `PROBer` vs. the full model. (B) `PROBer` vs. the `RSEM + PROBer*` pipeline. (C) `PROBer` vs. `StructureFold`, `Mod-seq`, and `icSHAPE`.

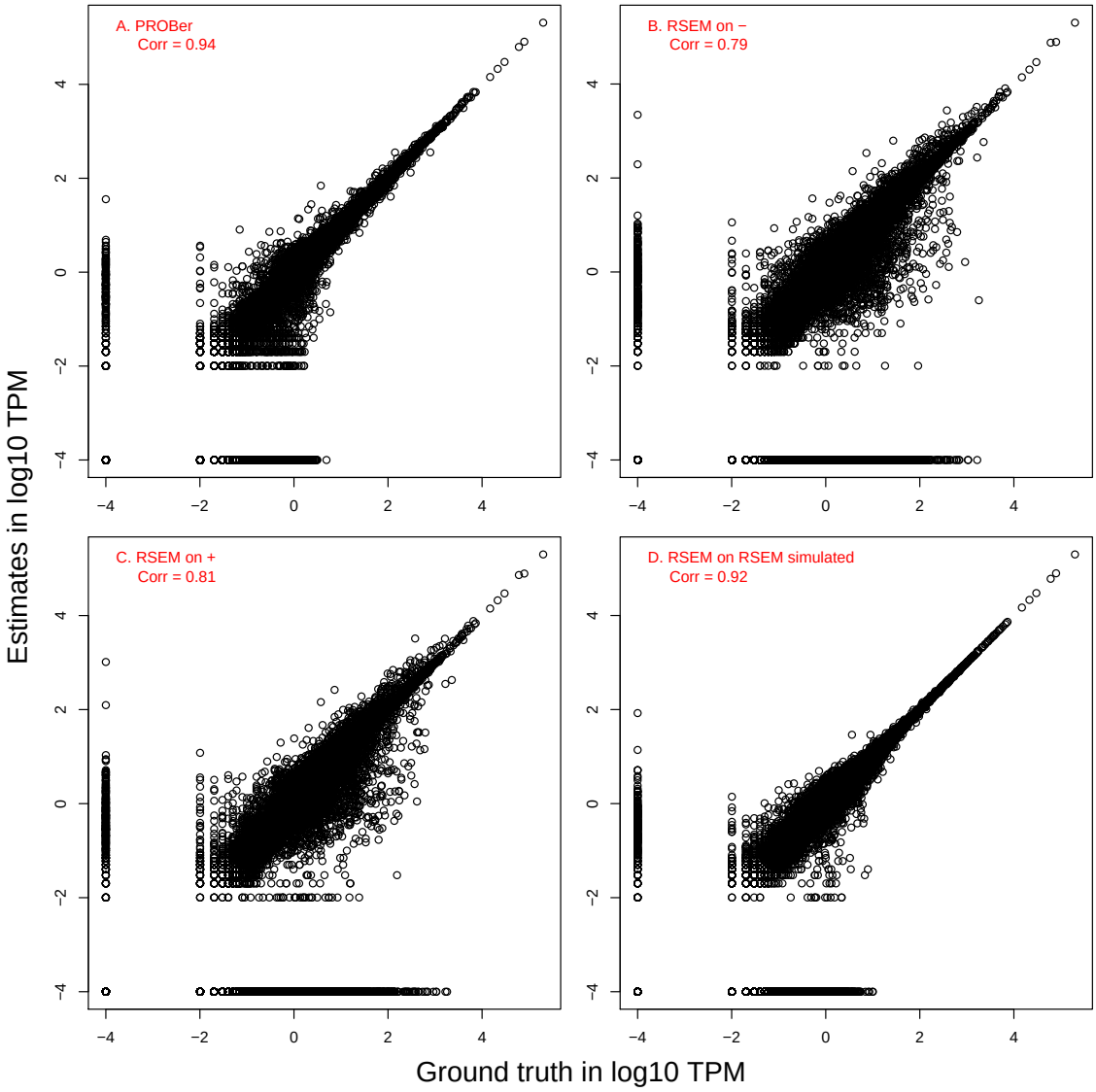

Figure S4. Related to Scattor plots showing RNA structural information improves transcript-level abundance estimation accuracy. In the plots, x-axis shows the ground truth abundances in log10 scale and y-axis shows the estimates in log10 scale. The four scatter plots are: A. `PROBer` estimates vs. ground truth. B. `RSEM` estimates on the control group vs. ground truth. C. `RSEM` estimates on the treatment group vs. ground truth. D. `RSEM` estimates on the `RSEM` simulated data set vs. ground truth. The log10 scaled Pearson's correlation coefficients between the estimates and ground truth are shown in the topleft corner. In order to avoid taking log10 of 0, 10^{-4} was added to all values. If we believe that RT random drop-off and RNA structure introduce biases to RNA-Seq, we'd better corret them during the quantification process. `RSEM` is not aware of either RT random drop-off or RNA secondary structure. Thus its performance drops when these biases are introduced (comparing B and C with D). When taking these biases into consideration, we can regain the performance (A).

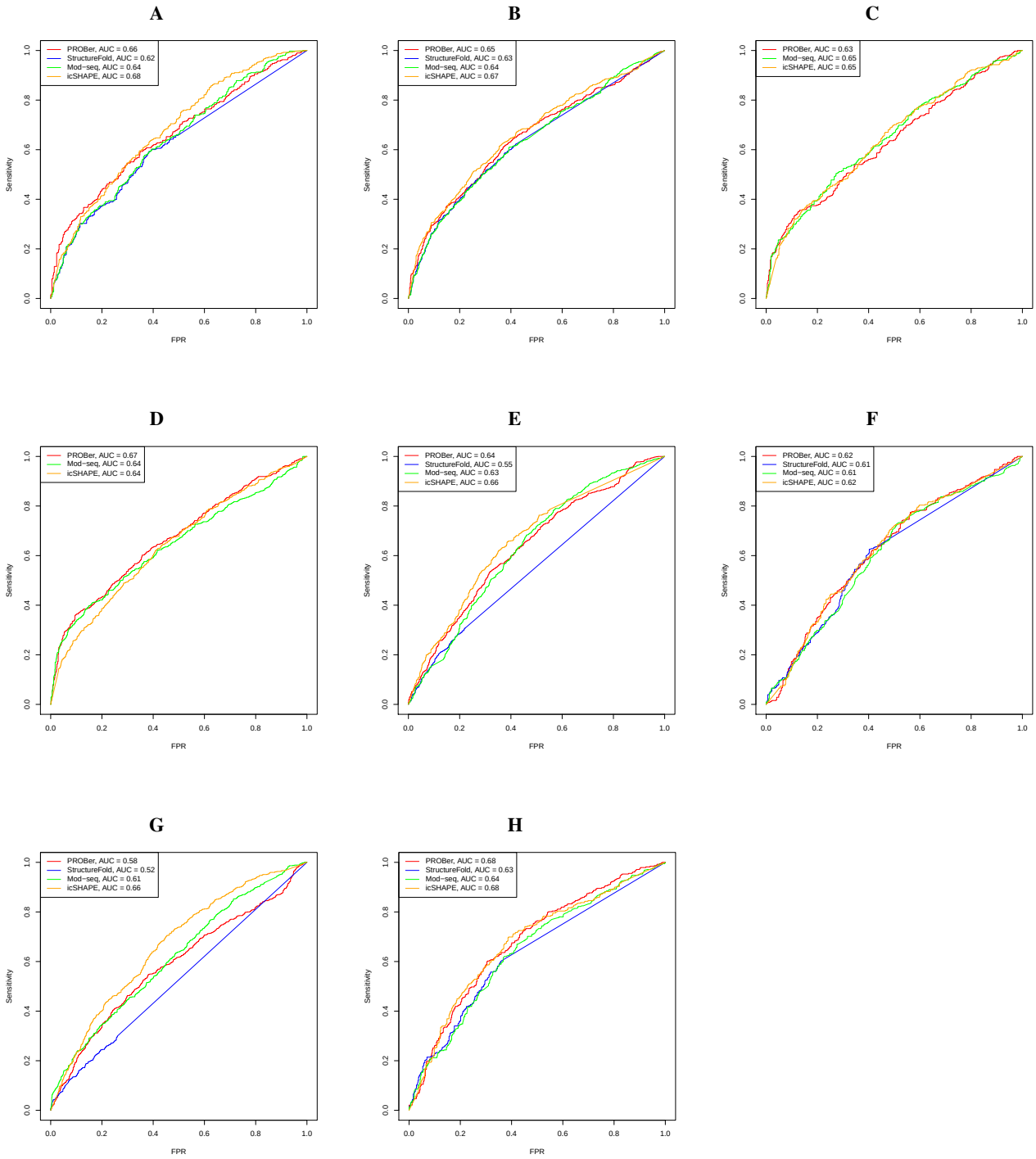

Figure S5. Related to Figure 2A. ROC curves for ribosomal RNAs on a variety of structure-probing data sets. `PROBer`, `StructureFold`, `Mod-seq`, and `icSHAPE` are evaluated and their corresponding area under curve (AUC) values are shown in the legend. In each plot, X-axis gives the False Positive Rate (FPR) and y-axis describes the True Positive Rate (sensitivity). (A) 18S rRNA and (B) 25S rRNA on Ding et al. Arabidopsis data. (C) 18S rRNA and (D) 25S rRNA on Talkish et al. yeast data. *In vitro* (E) 18S rRNA and (F) 12S mitochonrial rRNA, and *in vivo* (G) 18S rRNA and (H) 12S mitochonrial rRNA on Spitale et al. mouse data. (A-D) Because the modification agent is DMS, only positions with 'A's and 'C's are considered. (C-D) `StructureFold` is not included because its 2%-8% normalization step produces a normalization factor of 0 and thus results in undefined behaviors. (E-H) All four nucleotides are included in the analysis. Because the structure of mouse 25S rRNA is not available, we picked mitochondrially encoded 12S rRNA (12S_Mt) instead. Mouse 12S_Mt rRNA is 955 nt long and has ground truth structure. (A-H) We excluded the last "read length minus one" nucleotides of each rRNA from the analysis because little reads aligned to 3' end of transcripts. The excluded lengths are 36 nt, 49 nt, and 86 nt for Ding et al., Talkish et al., and Spitale et al. data respectively.

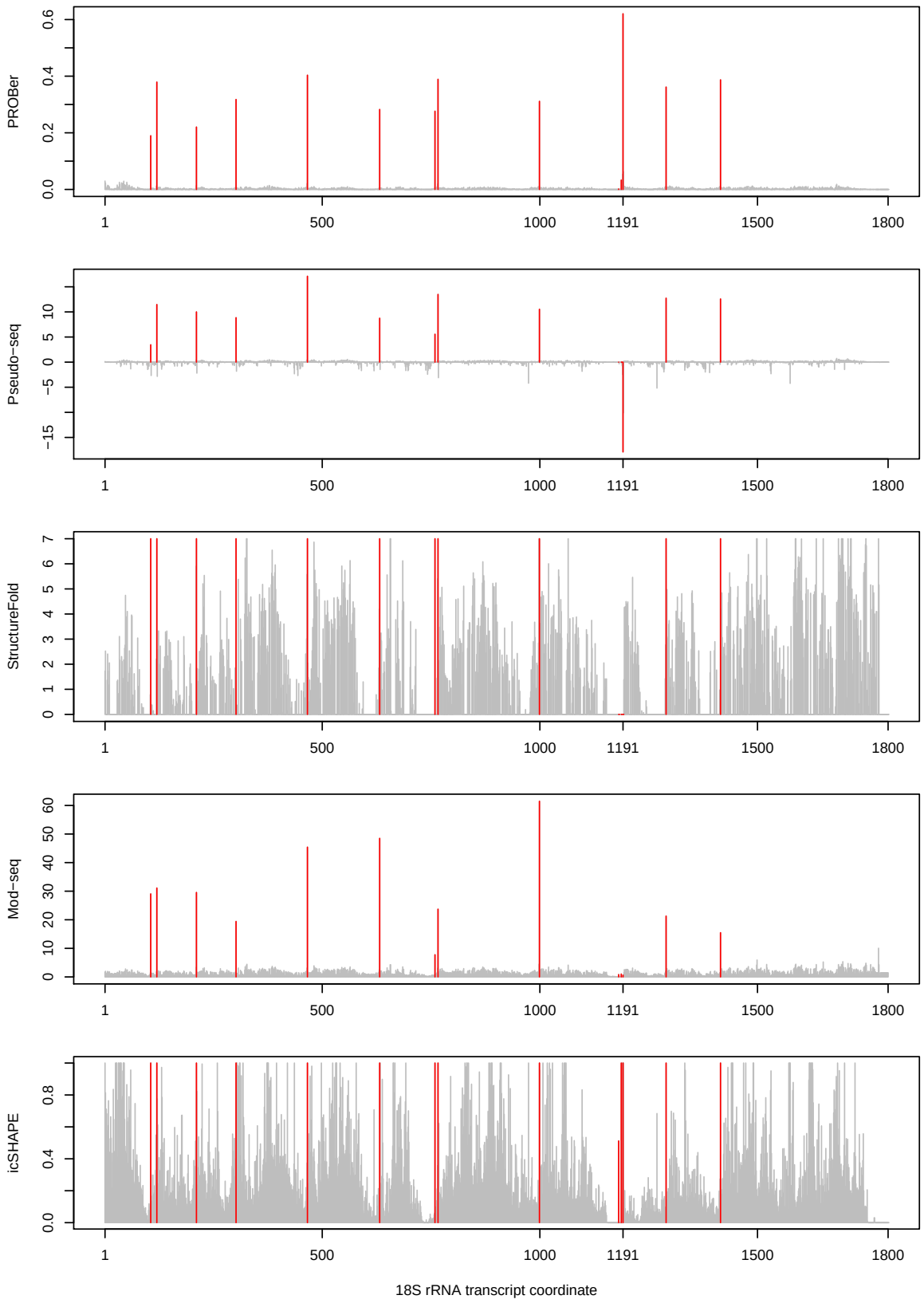

Figure S6. Related to Figure 2B. Only `PROBer` successfully detects m^{1}acp$^3\Psi$1191 in yeast 18S rRNA. We compared `PROBer` with `Pseudo-seq`, `StructureFold`, `Mod-seq`, and `icSHAPE`. Each row of the plot plots one method's score against the 18S rRNA transcript coordinates. Ψ sites are highlighted with red color and all other sites are plotted as grey. We can see that only `PROBer` detects m^{1}acp$^3\Psi$1191 at position 1191. `Pseudo-seq` gives a negative signal and `StructureFold` & `Mod-seq` give no signal at this site. Although `icSHAPE` gives a strong signal, it contains too many false positives.

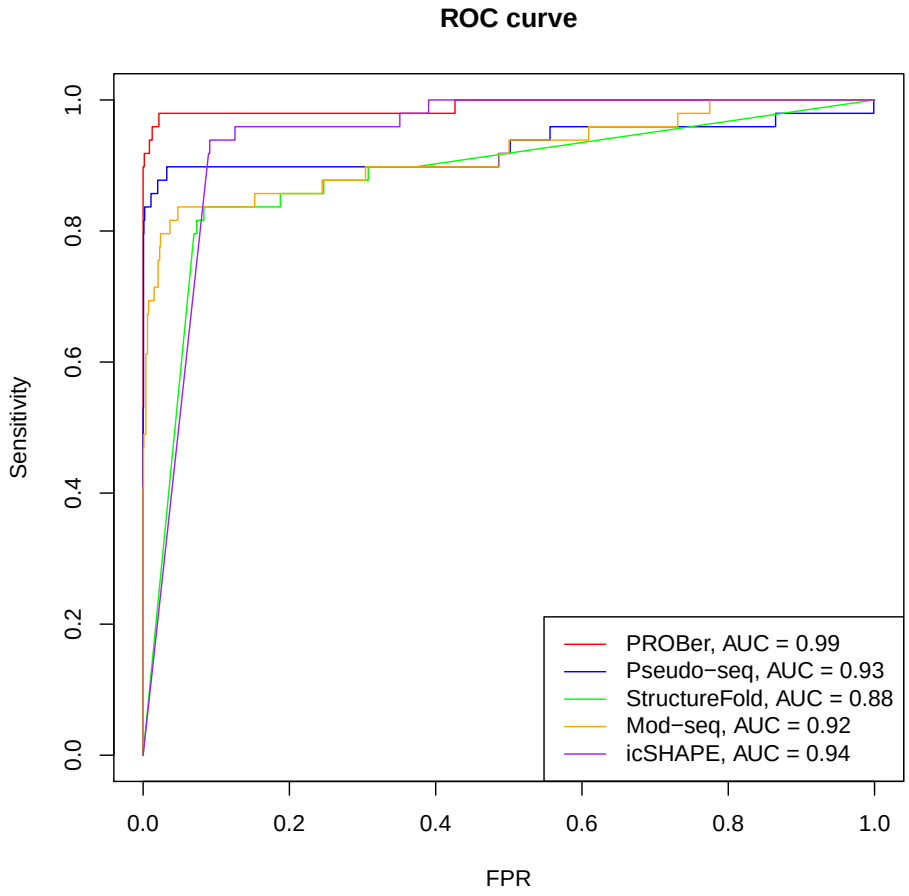

Figure S7. Related to Figure 2B and Experimental Procedures. ROC curves for all known Ψ sites in rRNAs and snoRNA on Carlile et al. Pseudo-seq data. X-axis gives the False Positive Rate (FPR) and y-axis describes the True Positive Rate (sensitivity). `PROBer`, `Pseudo-seq`, `StructureFold`, `Mod-seq`, and `icSHAPE` are evaluated and their corresponding area under curve (AUC) values are shown in the legend. We can observe that 1) `PROBer` outperforms other methods; 2) all methods have decent performance.

		Unique	Multi-mapping	Unalignable	Filtered
total RNA	+	88,979,408	20,475,607	7,786,678	602
	-	61,700,225	12,649,949	7,245,022	1,154
rRNA-	+	34,167,445	20,382,566	7,786,678	602
	-	23,133,282	12,528,974	7,245,022	1,154

Table S1. Related to Figure 1B. Alignment statistics of the modification-treated (+) and mock-treated (−) experiments for Ding et al. Arabidopsis data. The four columns represent the number of reads aligned to a unique transcript, multiple transcripts, no transcript, and reads that were filtered due to their having more than 200 ambigious alignments. We included all RNAs described in the online methods for calculating the numbers in the rows marked 'total RNA'. For rows marked 'rRNA-', we excluded reads aligned to ribosome RNAs. We can find that multi-mapping reads compose a significant portion of the total reads. In particular, when ribosome-derived reads are removed, more than a third of alignable reads are multi-mapping.

Unique	Multi-mapping	Unalignable	Filtered
8,033,885	2,395,537	6,808,057	1,486,909

Table S2. Related to Figure 1B. Alignment statistics of the pre-processed RBFOX2 iCLIP data set from Nostrand et al. The four columns represent the number of uniquely-mapping, multi-mapping, unalignable and filtered reads. Reads with more than 100 ambigious alignments were filtered.

RNA \ TPM	100	1000	10,000	100,000
RNase P	**0.63**/0.14/0.17/0.35	**0.98**/0.24/0.18/0.47	**0.99**/0.22/0.17/0.46	**1.00**/0.27/0.20/0.47
pT181 long	**0.38**/0.30/0.31/0.36	**0.94**/0.42/0.23/0.60	**0.96**/0.47/0.13/0.59	**0.98**/0.48/0.25/0.60
pT181 short	0.16/0.28/0.34/**0.41**	0.37/0.36/0.17/**0.53**	**0.69**/0.42/0.22/0.56	**0.84**/0.37/0.23/0.55

Table S3. Related to Figure 2A. Digital spike-in results. Each row lists a different RNA molecule and each column lists a different expression level. For a particular expression level, we assign each of the three RNA molecues that expression level in the corresponding simulated data set. In each cell, we list four Pearson's correlation coefficients in the order of `PROBer`, `StructureFold`, `Mod-seq`, and `icSHAPE`. The highest value in each cell is marked as bold. We excluded the last 36 nt of each RNA from the analysis. In addition, we used all 4 nucleotides instead of just 'A' and 'C' because the SHAPE agent can modify all 4 nucleotides.

Supplemental Experimental Procedures

1 Overview

Post-transcriptional regulation of gene expression plays a key role in many biological processes. This regulation can be understood from several perspectives, such as RNA secondary and tertiary structures, post-transcriptional modification of RNA nucleotides, and RNA-protein interactions. Recent advances in massively parallel DNA sequencing have enabled us to investigate each of these facets at the transcriptome scale through a diverse set of toeprinting assays, such as DMS/SHAPE-Seq (Ding et al., 2014; Rouskin et al., 2014; Spitale et al., 2015; Talkish et al., 2014), Pseudo-seq (Carlile et al., 2014) and iCLIP (König et al., 2010).

These toeprinting assays share a common workflow (shown in Figure 1A): chemically modifying RNAs to encode signals of interest, decoding these chemical marks by reverse transcriptase drop-off, and lastly, sequencing and mapping the resulting cDNA toeprints to recover the chemical modification "signatures". As such, these assays also share common analyses that need to be addressed in a unified manner.

We present PROBer, the first principled and unified framework for analyzing transcriptome-wide sequencing-based toeprinting assays. Our model is inspired by the RNA-Seq models of (Bray et al., 2016; Li et al., 2010; Li and Dewey, 2011; Roberts and Pachter, 2013; Trapnell et al., 2010) modified and extended to incorporate the steps unique to these toeprinting experiments.

We assume that our toeprinting experiment consists of two experimental groups: treatment and control. In the treatment group, RNA molecules are chemically modified followed by preparation of a sequencing library that is constructed using fragmentation & adapter ligation / random priming, PCR amplification and size selection. In the control group, the RNA molecules are not modified but all other library preparation steps are the same as the treatment group. The control group is used to measure and account for natural reverse transcriptase (RT) drop-off, random primer collision and other experimental biases. We denote the treatment and control groups by $+$ and $-$.

Given data from the treatment and control groups, our goal is to infer the modified nucleotide positions and their associated modification intensities across the whole transcriptome. We face two major challenges: to distinguish modification signal from drop-off noise, and to appropriately allocate reads that map to multiple locations. There has been previous work on both of these problems (in isolation). Existing methodology for SHAPE-Seq experiments has shown that modification signal can be distinguished from natural RT drop-off for single transcripts (Aviran et al., 2011a; Aviran and Pachter, 2014; Aviran et al., 2011b). The second challenge is fundamental to RNA-Seq, and methods for its solution have been developed in (Bray et al., 2016; Li et al., 2010; Li and Dewey, 2011; Roberts and Pachter, 2013; Trapnell et al., 2010). Our model combines and extends ideas from all of these methods in order to quantify toeprinting information on a transcriptome scale; in this sense it can be viewed as a generalization of these previous works.

Note that the iCLIP protocol (König et al., 2010) only has the treatment group. Thus, we cannot distinguish between real signal and RT drop-off noise from iCLIP data sets. In addition, a majority of iCLIP reads align to introns and intergenetic regions (König et al., 2010) and therefore we need map iCLIP reads to the genome, instead of a set of transcript sequences. Because of these reasons, PROBer handle iCLIP data differently than other toeprinting data. We will discuss how PROBer processes iCLIP data in Section 5. For now, let us focus on toeprinting assays that have controls and align their reads to transcript references, which include both RNA strcuture (Ding et al., 2014; Rouskin et al., 2014; Spitale et al., 2015; Talkish et al., 2014) and RNA modification (Carlile et al., 2014) assays.

There are two ways to initiate reverse transcription: transcript fragmentation & adapter ligation, and random priming. Because these two ways have little difference in our probabilistic model, without loss of generality, we assume random priming is used. Differences in modeling fragmentation-based protocols will be note when it is necessary. In order to explain our model and software, we begin with some background and notation:

1.1 Transcriptome abundance

We assume the reference transcript sequences are known. Suppose there are M transcripts in total, numbered from 1 to M. We use $\mathscr{L} = (\ell_1, \ldots, \ell_M)$, and $S = (s_1, \ldots, s_M)$ to denote the lengths and sequences of the transcripts. In addition, we define a special "transcript" — transcript 0, which is useful for modeling reads that are not compatible with the reference transcripts. We use $T = (\rho_1, \ldots, \rho_M)$ to denote the relative abundances of transcripts in the transcriptome.

T satisfies $\sum_{i=1}^{M} \rho_i = 1$ and $\text{TPM}_i = \rho_i \cdot 10^6$, where TPM stands for **T**ranscripts **P**er **M**illion and is a relative unit for expression levels.

To model the read generating process, we define the read generating probability vector, $A = (\alpha_0, \alpha_1, \ldots, \alpha_M)$, which represents the probabilities that a read is generated from either background noise (α_0) or any of the reference transcripts ($\alpha_i, i > 0$). The vector A satisfies $\sum_{i=0}^{M} \alpha_i = 1$. We further assume that reads are sequenced from transcripts at a rate proportional to the product of transcript abundance and length. Under this assumption, standard in RNA-Seq modeling, the relationship between T and A is given by

$$\rho_i = \frac{\alpha_i/(\ell_i - l_p + 1)}{\sum_{k=1}^{M} \alpha_k/(\ell_k - l_p + 1)}, \quad i = 1, \ldots, M, \tag{1}$$

where l_p is the length of random primer.

In the above equation, $\ell_i - l_p + 1$ is the *effective* length of transcript i, which can be thought as the number of transcript positions that a random primer can bind. Note that in fragmentation-based protocols, $l_p = 0$ and *effective* length is the same as transcript length.

In our model, many notations, parameters and formulae in the two experimental groups are exactly the same except that they are in different groups. In this article, we use $+$ and $-$ signs in the superscripts to distringuish parameters in different groups when it is necessary. Otherwise, we will ignore the $+$ and $-$ signs to show the common things they share. For example, A mentioned above represents a read generating probability vector. If we use A^+ and A^-, we refer to the read generating probability vectors in the treatment and control groups respectively.

To reduce the number of parameters contained in our model, we assume the relative transcript abundances are exactly the same in treatment and control groups, which we denote as T. Given T and the background noise probabilites, α_0^+ and α_0^-, from two experimental groups, we can recover the read generating probability vectors by the following formulae:

$$\begin{cases} \alpha_i^+ = (1 - \alpha_0^+) \cdot \frac{\rho_i(\ell_i - l_p + 1)}{\sum_{k=1}^{M} \rho_k(\ell_k - l_p + 1)} \\ \alpha_i^- = (1 - \alpha_0^-) \cdot \frac{\rho_i(\ell_i - l_p + 1)}{\sum_{k=1}^{M} \rho_k(\ell_k - l_p + 1)} \end{cases}, \quad i = 1, \ldots, M. \tag{2}$$

1.2 Toeprinting parameters

For each transcript, we number its positions from 5' end to 3' end, beginning with 1. We then define two sets of rates across all transcript and position combinations: chemical modification rates, B, and background reverse transcription stop rates, Γ. The chemical modification rates, B, are the signals of interest.

$$\begin{aligned} B &= \{\beta_{i,j} \mid 0 \le \beta_{i,j} \le 1\}, \\ \Gamma &= \{\gamma_{i,j} \mid 0 \le \gamma_{i,j} \le 1\}. \end{aligned}$$

For position j of transcript i, $\beta_{i,j}$ is the probability that the position is chemically modified and $\gamma_{i,j}$ is the probability that reverse transcription stops one nucleotide away from this position due to either natural RT drop-off or other background noise. For simplicity, we assume $\beta_{i,0} = \gamma_{i,0} = 1$ for all $i > 0$.

We further assume that the events of chemical modification and background reverse transcription stop are independent. This means both that the same type of events (e.g. chemical modification) are independent across different transcripts and positions, and different types of events (e.g. chemical modification and background reverse transcription drop-off) are independent at the same transcript position. This assumption is consistent with previous modeling of SHAPE-Seq (Aviran et al., 2011a; Aviran and Pachter, 2014; Aviran et al., 2011b).

1.3 Sequencing reads

We assume that we have sequenced N^+ and N^- reads in the treatment and control groups respectively. We also assume that the reads are strand-specific and in particular, single-end reads (or first mates of paired-end reads) are sequenced

from the forward strand of transcripts. For paired-end reads, the second mates are sequenced from the reverse strand. Note that strand-specificity is a characteristic of toeprinting protocols (Carlile et al., 2014; Ding et al., 2014; König et al., 2010; Rouskin et al., 2014; Spitale et al., 2015; Talkish et al., 2014). For single-end reads, we further assume that all reads have the same read length, L. Single-end reads only contain information about where an RT stops but not where it starts. However, if a transcribed fragment is shorter than L, all its information will be recorded in a single-end read. In this case, we just trim off the extra bases and treat the trimmed read as a full fragment.

Our model addresses four types of sequencing errors: substitution, insertion and deletion, and also "catastrophic" error resulting in reads from background noise. We use $\mathscr{E}$ to denote these sequencing error parameters:

$$\mathscr{E} = \{w_p(r_b, s_b), w_q(r_b, s_b), P_{trans}(n|c), P_{init}(s), P_I(b), P_{noise}(b)\},$$

where $w_p(r_b, s_b), w_q(r_b, s_b)$ are used for substitutions, $P_{trans}(n|c), P_{init}(s), P_I(b)$ are used for insertions and deletions, and $P_{noise}(b)$ is used for "catastrophic" reads.

Note that treatment and control groups each have their own set of sequencing error parameters, which we denote as $\mathscr{E}^+$ and $\mathscr{E}^-$. The parameters in $\mathscr{E}^+$ and $\mathscr{E}^-$ are the same but their values can be different.

Substitution. We model substitution errors by a series of position-specific or quality-score-specific substitution matrices (Li et al., 2010; Li and Dewey, 2011). We define the position-specific substitution matrices as

$$w_p(r_b, s_b) = P(r_b|s_b, p), \quad \text{with} \quad \sum_{r_b \in \{A,C,G,T,N\}} w_p(r_b, s_b) = 1.$$

$w_p(r_b, s_b)$ gives the conditional probability of observing base r_b at read position p given its aligned reference base is s_b. We define the quality-score-specific substitution matrices as

$$w_q(r_b, s_b) = P(r_b|s_b, q), \quad \text{with} \quad \sum_{r_b \in \{A,C,G,T,N\}} w_q(r_b, s_b) = 1.$$

$w_q(r_b, s_b)$ gives the conditional probability of observing read base r_b given its Phred quality score q and aligned reference base s_b.

Insertion and deletion. We model insertion and deletion errors using a first order Markov chain with three states: Insertion (I), Deletion (D), and Match (M). An insertion means a read base is inserted. A deletion means a reference base is deleted. A match means a read position aligns to a reference position, but that bases can be either a match or a mismatch. We further define: 1) the transition matrix $P_{trans}(x|c)$, which provides the conditional probability of next state x given current state c; 2) the initial state vector $P_{init}(s)$, which gives the probabilities of the first state s between a read and a reference sequence; 3) the base insertion probability vector $P_I(b)$, which determines the insertion probability of each base given an insertion state. $P_{trans}(x|c)$, $P_{init}(s)$, and $P_I(b)$ satisfy the constraints:

$$\sum_{x \in \{I,D,M\}} P_{trans}(x|c) = 1,$$
$$\sum_{s \in \{I,D,M\}} P_{init}(s) = 1,$$
$$\sum_{b \in \{A,C,G,T,N\}} P_I(b) = 1.$$

"Catastrophic" reads. These reads are generated from the background noise (α_0). For "catastrophic" reads, we assume that each base is independently generated from $P_{noise}(b)$, where

$$\sum_{b \in \{A,C,T,G,N\}} P_{noise}(b) = 1.$$

Lastly, we define Ξ as the set of all parameters of our model:

$$\Xi = \{T, \alpha_0^+, \alpha_0^-, B, \Gamma, \mathscr{E}^+, \mathscr{E}^-\}. \tag{3}$$

2 Probabilistic generative model

Figure S1A shows our probabilistic generative model. In this model, the expression levels, T, and background reverse transcription stop rates, Γ, are shared between the treatment and control groups. Because the read generating processes in the two groups are very similar, we will focus on the treatment group for explaining our model.

To generate a read, first $I^+ = i$, the isoform from which the read originates, is chosen based on A^+. A^+ can be calculated from T and α_0^+ by (2). Provided $i > 0$, $F^+ = f$, the fragment to which the read belongs is selected. A fragment is determined by its priming site j and length l, i.e. $f = (j,l)$. To generate a fragment, first the site j where a primer anneals is selected uniformly. Then the primer is extended toward the 5' end until RT stops. The resulting fragment length is l and we use $P(l|i,j)$ to denote the probability that a primer extends to length l given that it anneals to position j of isoform i. Next the fragment length is evaluated according to a size selection. If $l_{\min} \leq l \leq l_{\max}$, $P^+ = p = 1$, and the fragment passes size selection. Otherwise, $p = 0$. If $p = 1$, an observed read $R^+ = r$ is generated according to $\mathscr{E}^+$. If $p = 0$, no observed read is generated. In other words,

$$\begin{aligned} P(i,f,p,r) &= P(i|A^+)P(f|i)P(p|f)P(r|i,f,p,\mathscr{E}^+), \\ &= \alpha_i^+ \frac{1}{\ell_i - l_p + 1} P(l|i,j)P(p|l)P(r|i,j,l,p,\mathscr{E}^+), \\ \text{where} \quad P(p|l) &= \begin{cases} 1, & p = 0 \text{ and } (l < l_{\min} \text{ or } l > l_{\max}) \text{ or} \\ & p = 1 \text{ and } l_{\min} \leq l \leq l_{\max} \\ 0, & \text{otherwise} \end{cases} . \end{aligned}$$

The conditional probabilities can be described in terms of the parameters of the model. First, the conditional probability $P(l|i,j)$ is derived from the rules for primer extension. In the treatment group, reverse transcription stops because 1) RT hits a chemical modification or 2) RT drops off or hits a primer. If the fragment length is l, it must be the case that RT did not stop at positions $j-l+1,\ldots,j-l_p$ and stopped at position $j-l$. Thus for the treatment group, $P(l|i,j)$ is given by

$$P(l|i,j) = (1-(1-\beta_{i,j-l})(1-\gamma_{i,j-l})) \prod_{k=j-l+1}^{j-l_p} (1-\beta_{i,k})(1-\gamma_{i,k}).$$

In the control group, reverse transcription only stops due to reason 2). Thus for the control group, $P(l|i,j)$ is

$$P(l|i,j) = \gamma_{i,j-l} \prod_{k=j-l+1}^{j-l_p} (1-\gamma_{i,k}).$$

If a fragment passes the size selection, we need to generate an observed read from it. The observed variable, R^+, can either represent a single-end read or a paired-end read. Although in principle our model can handle a mixture of single-end and paired-end reads, here we assume the data consist of only a single type of reads. If single-end reads are sequenced, reads with length L are generated, unless the fragments are shorter than L. If paired-end reads are sequenced, our model estimates additional mate length distributions for first and second mates separately and the probability of generating r is the product of probabilities of picking particular mate lengths and probabilities of generating read bases given the mate lengths. The mate length distributions used here are similar to the ones proposed in (Li and Dewey, 2011) and thus the details are omitted.

For simplicity, we focus on generating single-end reads. In addition, we assume that each read comes with its Phred quality score sequence, q. q is generated from a first order Markov chain (Li and Dewey, 2011) and we denote its probability as $P(q)$. In this case, $R^+ = (r,q)$. If quality scores are not available, we just need to replace the quality-score-specific substitution matrices with the position-specific substitution matrices.

We first need to generate a sequence of hidden error states to describe where insertions and deletions occur. We denote this sequence of states as h. For example, if no insertion or deletion occurs, $h = \{M\}^L$ and its length $|h| = L$. We then define two mapping functions, $f_h^s(k)$ and $f_h^r(k)$, which map the k th hidden state to its aligned reference position (relative to the RT stop position $j-l$) and read position respectively. We calculate $f_h^s(k)$ and $f_h^r(k)$ by the following equations:

$$f_h^s(k) = \begin{cases} 0, & k = 0 \\ f_h^s(k-1)+1, & k > 0 \text{ and } h[k] = M \text{ or } D \\ f_h^s(k-1), & k > 0 \text{ and } h[k] = I \end{cases}, \tag{4}$$

$$f_h^r(k) = \begin{cases} 0, & k = 0 \\ f_h^r(k-1)+1, & k > 0 \text{ and } h[k] = M \text{ or } I \\ f_h^r(k-1), & k > 0 \text{ and } h[k] = D \end{cases}. \tag{5}$$

In the above equations, $[k]$ is used to extract the kth element in a sequence. Lastly, we define the probability of generating the corresponding read base at the kth hidden state as $w_h(k)$. We calculate $w_h(k)$ by

$$w_h(k) = \begin{cases} w_{q[f_h^r(k)]}(r[f_h^r(k)], s_i[j-l+f_h^s(k)]), & h[k] = M \\ P_I(r[f_h^r(k)]), & h[k] = I \\ 1.0, & h[k] = D \end{cases}.$$

Then the read generating probability is

$$P(r,q|i,j,l,p=1,\mathscr{E}) = P(q)\sum_h P_{init}(h[1]) \cdot \prod_{k=2}^{|h|} P_{trans}(h[k] \mid h[k-1]) \cdot \prod_{k=1}^{|h|} w_h(k).$$

The summation in the above equation is over all hidden sequences h that can generate the observed read r.

If a read is "catastrophic" (transcript 0), its bases are generated independently according to P_{noise}. The probability of generating a single-end "catastrophic" read with length L is

$$P(r|i=0) = \prod_{k=1}^{L} P_{noise}(r[k]).$$

We assume "catastrphic" reads are generated after the size selection. Thus we always have $p = 1$ if $i = 0$.

$$\underbrace{0,0,0}_{Y_1=3},1,\underbrace{0}_{Y_2=1},2,\underbrace{}_{Y_3=0},3,0,0,\ldots,N-1,\underbrace{0,0,0,0,0}_{Y_N=5},N$$

As illustrated in the above example, besides generating observed reads, our model also generates a different number of hidden fragments that fail to pass the size selection. We use random variable Y_i to represent the number of hidden fragments generated between observed reads $i-1$ and i. In addition, we use κ to represent the probability of generating either a "catastrophic" read or a fragment from the reference that passes the size selection:

$$\kappa = P(Z_0 = 1) + \sum_{\substack{i>0, \\ l_{\min} \le l \le l_{\max}}} P(Z_{ijl} = 1),$$

where Z_0 and Z_{ijl} are hidden indicator variables. $Z_0 = 1$ if and only if the fragment comes from the background noise. $Z_{ijl} = 1$ if and only if the fragment primes at position j of transcript i and has a length of l.

In our model, the event if a fragment passes the size selection is determined independently with a probability κ. Thus, Y_ns are independent and identically distributed (i.i.d.), and follow a geometric distribution, Geom(κ):

$$P(Y_n = k) = (1-\kappa)^k \kappa, \quad k = 0,1,2,\ldots.$$

Let us focus on $P_{obs}(R_n)$, the probability of observing the nth observed read with sequence R_n. We can calculate $P_{obs}(R_n)$ by summing over Y_n:

$$\begin{aligned} P_{obs}(R_n) &= \sum_{k=0}^{\infty} P(Y_n = k) P(R_n | Y_n = k), \\ &= (\sum_{k=0}^{\infty} P(Y_n = k)) \cdot \frac{P(R_n, P_n = 1)}{\kappa}, \\ &= 1 \cdot \frac{P(R_n)}{\kappa}, \\ &= \frac{P(R_n)}{\kappa}. \end{aligned}$$

In the above equations, $P(R_n|Y_n = k)$ is the conditional probability of observing read sequence R_n given that its fragment passes the size selection and $P(R_n|Y_n = k)$ is the same for all k. Therefore, $P(R_n|Y_n = k) = \frac{P(R_n, P_n=1)}{\kappa} = \frac{P(R_n)}{\kappa}$. To calculate $P(R_n)$, the marginal probability that the nth observed read passes the size selection and its observed sequence is R_n, we have to sum over a huge space of hidden states represented by indicator variables Z_{nijl1h}. $Z_{nijl1h} = 1$ suggests that the nth observed read comes from transcript i, starts at position j, has a fragment length of l, passes the size selection, and has a sequencing error state of h. Thus, we calculate $P(R_n)$ by

$$P(R_n) = \sum_{\substack{i=0 \text{ or }, \\ l_{\min} \le l \le l_{\max}}} P(Z_{nijl1h} = 1, R_n). \tag{6}$$

Now we are ready to write down the formula for the observed data likelihood, L_{obs}. If we denote the observed data in the treatment and control groups as $\mathscr{D}^+ = \{R_1^+, R_2^+, \cdots, R_{N^+}^+\}$ and $\mathscr{D}^- = \{R_1^-, R_2^-, \cdots, R_{N^-}^-\}$, L_{obs} is

$$\begin{aligned} L_{obs}(\Xi; \mathscr{D}^+, \mathscr{D}^-) &= \prod_{n=1}^{N^+} P_{obs}(R_n^+) \cdot \prod_{n=1}^{N^-} P_{obs}(R_n^-), \\ &= \prod_{n=1}^{N^+} \frac{P(R_n^+)}{\kappa^+} \cdot \prod_{n=1}^{N^-} \frac{P(R_n^-)}{\kappa^-}, \end{aligned} \tag{7}$$

where

$$\begin{aligned} \kappa^+ &= P(Z_0^+ = 1) + \sum_{\substack{i>0, \\ l_{\min} \le l \le l_{\max}}} P(Z_{ijl}^+ = 1), \\ &= \alpha_0^+ + \sum_{\substack{i>0, \\ l_{\min} \le l \le l_{\max}}} \alpha_i^+ \cdot \frac{1}{\ell_i - l_p + 1} \cdot (1 - (1 - \gamma_{i,j-l})(1 - \beta_{i,j-l})) \cdot \prod_{k=j-l+1}^{j-l_p} (1 - \gamma_{i,k})(1 - \beta_{i,k}), \qquad (8) \\ \kappa^- &= P(Z_0^- = 1) + \sum_{\substack{i>0, \\ l_{\min} \le l \le l_{\max}}} P(Z_{ijl}^- = 1) \\ &= \alpha_0^- + \sum_{\substack{i>0, \\ l_{\min} \le l \le l_{\max}}} \alpha_i^- \cdot \frac{1}{\ell_i - l_p + 1} \cdot \gamma_{i,j-l} \cdot \prod_{k=j-l+1}^{j-l_p} (1 - \gamma_{i,k}). \qquad (9) \end{aligned}$$

2.1 Alternative objective function

In practice, we found that using the objective function L_{obs} resulted in an unstable program with unexpected behaviors. For example, normally no reads align to the 3' end of transcripts and thus we have no way to obtain structural estimates at the 3' ends. However, the program using L_{obs} as its objective function reported very high β probabilities and very

low γ probabilities at the 3' end. After carefully investigation, we realized that interpolating the hidden reads increases estimators' variances.

Thus, we used the following objective function L_{alt}, which trades off some biases for a reduced variance:

$$L_{alt}(\Xi; \mathscr{D}^+, \mathscr{D}^-) = \prod_{n=1}^{N^+} P(R_n^+) \cdot \prod_{n=1}^{N^-} P(R_n^-). \tag{10}$$

L_{alt} is the data likelihood when we only consider observed reads.

3 Model inference via the Expectation-Maximization algorithm

We use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to learn our model parameters. We first describe how to learn model parameters under objective function L_{obs}. Because L_{obs} takes hidden reads into consideration, we call it the full model. The algorithmic workflow of full model is shown in Algorithm 1. In the end of this section, we will describe how `PROBer` learns model parameters under objective function L_{alt}.

Algorithm 1 EM algorithm for the full model

Input: treatment group data $\mathscr{D}^+$ and control group data $\mathscr{D}^-$.
Initialize model parameters Ξ ($= \{T, \alpha_0^+, \alpha_0^-, B, \Gamma, \mathscr{E}^+, \mathscr{E}^-\}$).
repeat
E step: Calculate expectations and collect summary statistics
1. Calculate $E(Y_{nijl}^+|\Xi)$ and $E(Z_{nijl1h}^+|\Xi, \mathscr{D}^+)$ from treatment group.
2. Calculate $E(Y_{nijl}^-|\Xi)$, and $E(Z_{nijl1h}^-|\Xi, \mathscr{D}^-)$ from control group.
3. Collect summary statistics: $X_i^+, C_{ik}^+, D_{ik}^+, X_i^-, C_{ik}^-, D_{ik}^-$ and error-model-related statistics such as $T_{cx}^+, W_{qr_bs_b}^+$.

M step: Re-estimate Ξ
1. Re-estimate $T, \alpha_0^+, \alpha_0^-$ based on X_i^+s and X_i^-s.
2. Re-estimate B, Γ based on C_{ik}^+s, D_{ik}^+s, C_{ik}^-s and D_{ik}^-s.
3. Re-estimate $\mathscr{E}^+, \mathscr{E}^-$ based on error-model-related summary statistics such as T_{cx}^+s and $W_{qr_bs_b}^+$s.

until Convergence
Output: Estimated Ξ.

In theory, an observed read r can come from any transcript with any starting position. Thus, in order to calculate the marginal probability $P(r)$, we have to sum over a huge space of hidden states, as demonstrated in (6). However, most hidden states contribute little to the marginal probability because the reference sequences implied by the hidden states would not match the read sequence. In order to reduce the computational cost, we align each observed read to the reference and use the resulting alignments to confine its hidden state space.

We define π to be the alignment-confined set of hidden states an observed read r can take. π consists of quads in the format of (i, j, l, h), which describe the transcript ID, start position, fragment length and hidden error state(s) the observed read r may have. We always assume that r can come from the background noise and thus $(0, 0, 0, \emptyset) \in \pi$. We approximate $P(r)$ by only summing over hidden states in π:

$$P(r) \approx \sum_{(i,j,l,h) \in \pi} P(Z_{ijl1h}, r). \tag{11}$$

Because aligners always return a set of high quality alignments, the approximation described in (11) captures most of the probability mass for $P(r)$.

If r is a paired-end read, the alignments have the format of (i, pos, l, h), where i is the transcript ID, pos is the aligned leftmost transcript position, l is the inferred fragment length, and h is the hidden sequencing error states for the two mates. Thus, for a paired-end read, its π becomes

$$\pi = \{(0, 0, 0, \emptyset)\} \cup \{(i, pos + l - 1, l, h) \mid (i, pos, l, h) \in \text{Alignments}\}.$$

If r is a single-end read, the alignments have the format of (i, pos, h), where i is the transcript ID, pos is the aligned leftmost transcript position, and h is the hidden sequencing error state. We have two cases to discuss. If r's read length $L_r < L$, it is trimmed and thus represents a full fragment. In this case, we augment its alignments from (i, pos, h) to (i, pos, L_r, h) and treat it as a "paired-end" read. Otherwise, it is not trimmed and has a read length of L. For each alignment (i, pos, h) it has, its fragment length can vary from $\max(l_{\min}, L)$ to $\min(l_{\max}, \ell_i - pos + 1)$. Thus, its π becomes

$$\pi = \{(0,0,0,\emptyset)\} \cup \{(i, pos+l-1, l, h) \mid \begin{array}{c} (i, pos, h) \in \text{Alignments}, \\ \max(l_{\min}, L) \le l \le \min(l_{\max}, \ell_i - pos + 1) \end{array} \}.$$

In the E step, we assume the model parameters are known. For each group, we are interested in calculating the expectations of two sets of hidden variables, Y_{nijl} and Z_{nijl1h}, given model parameters and observed data. Y_{nijl} represents the number of unobserved fragments that are generated between the $n-1$ and nth observed reads, prime at position j of transcript i, and have length l, where $l < l_{\min}$ or $l > l_{\max}$. We have defined Z_{nijl1h} in the last section.

We first focus on $E(Y_{nijl}|\Xi)$. As we discussed before, $Y_n \sim \text{Geom}(\kappa)$ and thus $E(Y_n) = \frac{1-\kappa}{\kappa}$. In addition, $E(Y_{nijl}|Y_n) = \frac{P(Z_{ijl}=1)}{1-\kappa} Y_n$. Thus, by the law of total expectation, $E(Y_{nijl}|\Xi)$ becomes

$$E(Y_{nijl}|\Xi) = E(E(Y_{nijl}|Y_n)) = E(\frac{P(Z_{ijl}=1)}{1-\kappa} Y_n) = \frac{P(Z_{ijl}=1)}{1-\kappa} \cdot \frac{1-\kappa}{\kappa} = \frac{P(Z_{ijl}=1)}{\kappa}.$$

For the treatment and control groups, we have

$$\begin{aligned} E(Y^+_{nijl}|\Xi) &= \begin{cases} \frac{P(Z^+_{ijl}=1)}{\kappa^+}, & l < l_{\min} \text{ or } l > l_{\max} \\ 0, & \text{otherwise} \end{cases}, \\ E(Y^-_{nijl}|\Xi) &= \begin{cases} \frac{P(Z^-_{ijl}=1)}{\kappa^-}, & l < l_{\min} \text{ or } l > l_{\max} \\ 0, & \text{otherwise} \end{cases}, \end{aligned}$$

where

$$\begin{aligned} P(Z^+_{ijl}=1) &= \alpha^+_i \cdot \frac{1}{\ell_i - l_p + 1} \cdot (1 - (1-\gamma_{i,j-l})(1-\beta_{i,j-l})) \cdot \prod_{k=j-l+1}^{j-l_p} (1-\gamma_{i,k})(1-\beta_{i,k}), \\ P(Z^-_{ijl}=1) &= \alpha^-_i \cdot \frac{1}{\ell_i - l_p + 1} \cdot \gamma_{i,j-l} \cdot \prod_{k=j-l+1}^{j-l_p} (1-\gamma_{i,k}). \end{aligned}$$

Calculating $E(Z_{nijl1h}|\Xi, \mathscr{D})$ is relatively easy. If $(i,j,l,h) \notin \pi_n$, $E(Z_{nijl1h}|\Xi, \mathscr{D}) = 0$. Otherwise, for each group, the expectation can be calculated as

$$\begin{aligned} E(Z^+_{nijl1h}|\Xi, \mathscr{D}^+) &= \frac{P(Z^+_{nijl1h}=1|\Xi) P(r^+_n | Z^+_{nijl1h}=1)}{\sum_{(i',j',l',h') \in \pi^+_n} P(Z^+_{ni'j'l'1h'}=1|\Xi) P(r^+_n | Z^+_{ni'j'l'1h'}=1)}, \\ E(Z^-_{nijl1h}|\Xi, \mathscr{D}^-) &= \frac{P(Z^-_{nijl1h}=1|\Xi) P(r^-_n | Z^-_{nijl1h}=1)}{\sum_{(i',j',l',h') \in \pi^-_n} P(Z^-_{ni'j'l'1h'}=1|\Xi) P(r^-_n | Z^-_{ni'j'l'1h'}=1)}. \end{aligned}$$

Let us define several summary statistics based on the previous expectations. These summary statistics will be used in the M step to re-estimate model parameters.

First, we define X^+_i and X^-_i as the number of expected fragments generated from transcript i in the two experimental groups. X^+_is and X^-_is are used to estimate transcript abundances. We calculate them by

$$
\begin{aligned}
X_i^+ &= \sum_{n=1}^{N^+}\Big(\sum_{j,l} E(Y_{nijl}^+|\Xi) + \sum_{j,l,h} E(Z_{nijl1h}^+|\Xi,\mathscr{D}^+)\Big),\\
X_i^- &= \sum_{n=1}^{N^-}\Big(\sum_{j,l} E(Y_{nijl}^-|\Xi) + \sum_{j,l,h} E(Z_{nijl1h}^-|\Xi,\mathscr{D}^-)\Big).
\end{aligned}
$$

Second, we define summary statistics for estimating B and Γ. Let C_{ik}^+ and D_{ik}^+ denote the expected number of fragments covering and dropping off at position k of transcript i in the treatment group. If a fragment's priming site $j \geq k + l_p$ and its leftmost base $j - l + 1 \leq k$, it covers k. If the fragment's leftmost base $j - l + 1 = k + 1$, it drops off at k. We calculate C_{ik}^+ and D_{ik}^+ by

$$
\begin{aligned}
C_{ik}^+ &= \sum_{n=1}^{N^+}\Big(\sum_{k+l_p\leq j\leq k+l-1} E(Y_{nijl}^+|\Xi) + \sum_{k+l_p\leq j\leq k+l-1} E(Z_{nijl1h}^+|\Xi,\mathscr{D}^+)\Big),\\
D_{ik}^+ &= \sum_{n=1}^{N^+}\Big(\sum_{l\leq \ell_i-k} E(Y_{n,i,k+l,l}^+|\Xi) + \sum_{l\leq \ell_i-k} E(Z_{n,i,k+l,l,1,h}^+|\Xi,\mathscr{D}^+)\Big).
\end{aligned}
$$

Similarly, let C_{ik}^- and D_{ik}^- denote the expected number of fragments covering and dropping off at position k of transcript i in the control group. We calculate them by

$$
\begin{aligned}
C_{ik}^- &= \sum_{n=1}^{N^-}\Big(\sum_{k+l_p\leq j\leq k+l-1} E(Y_{nijl}^-|\Xi) + \sum_{k+l_p\leq j\leq k+l-1} E(Z_{nijl1h}^-|\Xi,\mathscr{D}^-)\Big),\\
D_{ik}^- &= \sum_{n=1}^{N^-}\Big(\sum_{l\leq \ell_i-k} E(Y_{n,i,k+l,l}^-|\Xi) + \sum_{l\leq \ell_i-k} E(Z_{n,i,k+l,l,1,h}^-|\Xi,\mathscr{D}^-)\Big).
\end{aligned}
$$

Lastly, we define summary statistics used to estimate $\mathscr{E}$. We only pick two examples as a demonstration. We define T_{cx}^+ as the expected number of transitions from c to x among the observed reads in the treatment group. It is used to estimate the transition matrix $P_{trans}^+(x|c)$. We collect T_{cx}^+ by

$$
T_{cx}^+ = \sum_{n=1}^{N^+}\sum_{i>0,(i,j,l,h)\in\pi_n^+} E(Z_{nijl1h}^+|\Xi,\mathscr{D}^+)\sum_{k=1}^{|h|-1}\mathbf{1}_{cx}(h[k],h[k+1]),
$$

where $\mathbf{1}_{cx}$ is an indicator function and defined by

$$
\mathbf{1}_{cx}(a,b) = \begin{cases} 1, & a = c \text{ and } b = x \\ 0, & \text{otherwise} \end{cases}.
$$

In addition, we define $W_{qr_bs_b}^+$ as the expected number of occurences in the treatment group that reference base s_b generates a read base r_b under a quality score value q. $W_{qr_bs_b}^+$ is used to estimate quality-score-based subsititution matrices $w_q(r_b,s_b)$ and can be collected by

$$
W_{qr_bs_b}^+ = \sum_{n=1}^{N^+}\sum_{i>0,(i,j,l,h)\in\pi_n^+} E(Z_{nijl1h}^+|\Xi,\mathscr{D}^+)\sum_{k=1}^{|h|}\mathbf{1}_{qr_bs_b}(h[k],q_n[f_h^r(k)],r_n[f_h^r(k)],s_i[j-l+f_h^s(k)]),
$$

where $f_h^s(k)$ and $f_h^r(k)$ are defined by (4) and (5) and the indicator function $\mathbf{1}_{qr_bs_b}$ is defined as

$$
\mathbf{1}_{qr_bs_b}(h,q',a,b) = \begin{cases} 1, & h = M, q' = q, a = r_b, \text{ and } b = s_b \\ 0, & \text{otherwise} \end{cases}.
$$

Note that Y^+_{nijl} is not involved in the calculation of T^+_{cx} and $W^+_{qr_bs_b}$. Unobserved fragments are never sequenced and thus unrelated to sequencing errors.

In the M step, we re-estimate Ξ based on the collected summary statistics.

First, we re-estimate abundance related parameters, $T, \alpha^+_0, \alpha^-_0$, based on X^+_is and X^-_is:

$$\begin{aligned}
\hat{\rho}_i &= \frac{(X^+_i + X^-_i)/(\ell_i - l_p + 1)}{\sum_{k=1}^{M}(X^+_k + X^-_k)/(\ell_k - l_p + 1)}, \quad i = 1, \ldots, M, \\
\hat{\alpha}^+_0 &= \frac{X^+_0}{\sum_{k=0}^{M} X^+_k}, \\
\hat{\alpha}^-_0 &= \frac{X^-_0}{\sum_{k=0}^{M} X^-_k}.
\end{aligned}$$

Second, we re-estimate toeprinting parameters, B and Γ, based on C^+_{ik}s, D^+_{ik}s, C^-_{ik}s and D^-_{ik}s. We will discuss how to estimate these parameters in the next section.

Lastly, we re-estimate sequencing error related parameters, $\mathscr{E}^+$ and $\mathscr{E}^-$, based on sequencing-error-related summary statistics. For example, we re-estimate the transition matrix $P^+_{trans}(x|c)$ by

$$\hat{P}^+_{trans}(x|c) = \frac{T^+_{cx}}{\sum_{x' in \{I,D,M\}} T^+_{cx'}},$$

and the quality-score-based substitution matrices $w^+_q(r_b, s_b)$ by

$$\hat{w}^+_q(r_b, s_b) = \frac{W^+_{qr_bs_b}}{\sum_{r'_b \in \{A,C,G,T,N\}} W^+_{qr'_bs_b}}.$$

For computational efficiency, we only re-estimate $\mathscr{E}^+$ and $\mathscr{E}^-$ for the first 10 EM iterations.

3.1 PROBer's EM algorithm

Algorithm 2 PROBer's EM algorithm

Input: treatment group data $\mathscr{D}^+$ and control group data $\mathscr{D}^-$.
Initialize model parameters Ξ ($= \{T, \alpha^+_0, \alpha^-_0, B, \Gamma, \mathscr{E}^+, \mathscr{E}^-\}$).
repeat
E step: Calculate expectations and collect summary statistics
1. Calculate $E(Z^+_{nijl1h}|\Xi, \mathscr{D}^+)$ from treatment group.
2. Calculate $E(Z^-_{nijl1h}|\Xi, \mathscr{D}^-)$ from control group.
3. Collect summary statistics: $X^+_i, C^+_{ik}, D^+_{ik}, X^-_i, C^-_{ik}, D^-_{ik}$ and error-model-related statistics such as $T^+_{cx}, W^+_{qr_bs_b}$.

M step: Re-estimate Ξ
1. Re-estimate $T, \alpha^+_0, \alpha^-_0$ based on X^+_is and X^-_is.
2. Re-estimate B, Γ based on C^+_{ik}s, D^+_{ik}s, C^-_{ik}s and D^-_{ik}s.
3. Re-estimate $\mathscr{E}^+, \mathscr{E}^-$ based on error-model-related summary statistics such as T^+_{cx}s and $W^+_{qr_bs_b}$s.

until Convergence
Output: Estimated Ξ.

PROBer's algorithmic workflow is shown in Algorithm 2. We removed interpolating hidden reads from the EM algorithm. In addition, we redefine X^+_i, X^-_i, C^+_{ik}, D^+_{ik}, C^-_{ik}, and D^-_{ik} as:

$$
\begin{aligned}
X_i^+ &= \sum_{n=1}^{N^+} \sum_{j,l,h} E(Z^+_{nijl1h}|\Xi,\mathscr{D}^+),\\
X_i^- &= \sum_{n=1}^{N^-} \sum_{j,l,h} E(Z^-_{nijl1h}|\Xi,\mathscr{D}^-),\\
C_{ik}^+ &= \sum_{n=1}^{N^+} \sum_{k+l_p\le j\le k+l-1} E(Z^+_{nijl1h}|\Xi,\mathscr{D}^+),\\
D_{ik}^+ &= \sum_{n=1}^{N^+} \sum_{l\le \ell_i-k} E(Z^+_{n,i,k+l,l,1,h}|\Xi,\mathscr{D}^+),\\
C_{ik}^- &= \sum_{n=1}^{N^-} \sum_{k+l_p\le j\le k+l-1} E(Z^-_{nijl1h}|\Xi,\mathscr{D}^-),\\
D_{ik}^- &= \sum_{n=1}^{N^-} \sum_{l\le \ell_i-k} E(Z^-_{n,i,k+l,l,1,h}|\Xi,\mathscr{D}^-).
\end{aligned}
$$

4 Maximum a posteriori estimates of toeprinting parameters

This section discusses how to obtain Maximum a posteriori (MAP) estimates of toeprinting parameters, β_{ik} and γ_{ik}, based on summary statistics $C_{ik}^+, D_{ik}^+, C_{ik}^-, D_{ik}^-$ for all i,k. We do not use maximum likelihood (ML) estimates because we cannot obtain reliable ML estimates for most toeprinting parameters from these transcriptome-wide assays due to limited sequencing depth.

We introduce Beta distributions as priors for every $\beta \in B$ and $\gamma \in \Gamma$ because the Beta distribution is conjugate to the binomial distribution. The priors of β_{ik} and γ_{ik} are parameterized as

$$
\begin{aligned}
\beta_{ik} &\sim Beta(1+d_{\beta_{ik}}, 1+c_{\beta_{ik}}), \quad \text{with} \quad d_{\beta_{ik}}>0, c_{\beta_{ik}}>0,\\
\gamma_{ik} &\sim Beta(1+d_{\gamma_{ik}}, 1+c_{\gamma_{ik}}), \quad \text{with} \quad d_{\gamma_{ik}}>0, c_{\gamma_{ik}}>0.
\end{aligned}
$$

For simplicity, we focus on MAP estimates at position k of transcript i and ignore the subscripts. The log joint probability function, which takes the prior terms into consideration, is

$$
\begin{aligned}
l_{\text{MAP}} &= (c_\gamma \log(1-\gamma) + d_\gamma \log\gamma + C^- \log(1-\gamma) + D_i^- \log\gamma) +\\
&\quad (c_\beta \log(1-\beta) + d_\beta \log\beta + C^+ \log(1-\gamma)(1-\beta) + D^+ \log(1-(1-\gamma)(1-\beta)).
\end{aligned} \tag{12}
$$

Note that we omit the normalization constants for the Beta distributions in l_{MAP} because these constants have no effects on the MAP estimates.

Differentiate l_{MAP} with respect to γ, β and set the derivatives to 0:

$$
-\frac{c_\gamma + C^-}{1-\gamma} + \frac{d_\gamma + D^-}{\gamma} - \frac{C^+}{1-\gamma} + \frac{D^+(1-\beta)}{\gamma+\beta-\gamma\beta} = 0, \tag{13}
$$

$$
-\frac{c_\beta}{1-\beta} + \frac{d_\beta}{\beta} - \frac{C^+}{1-\beta} + \frac{D^+(1-\gamma)}{\gamma+\beta-\gamma\beta} = 0. \tag{14}
$$

Substracting $\frac{1-\beta}{1-\gamma} \cdot (14)$ from (13), we obtain

$$
-\frac{c_\gamma + C^- - c_\beta + d_\beta \frac{1-\beta}{\beta}}{1-\gamma} + \frac{d_\gamma + D^-}{\gamma} = 0. \tag{15}
$$

Rearranging (15), we have

$$\gamma = \frac{d_\gamma + D^-}{c_\gamma + C^- + d_\gamma + D^- + (d_\beta \frac{1-\beta}{\beta} - c_\beta)}. \tag{16}$$

Plugging (16) into (14), we obtain

$$-\frac{c_\beta + C^+}{1-\beta} + \frac{d_\beta}{\beta} + \frac{D^+(c_\gamma + C^- - c_\beta + d_\beta \frac{1-\beta}{\beta})}{\beta(c_\gamma + C^- - c_\beta - d_\beta) + (d_\beta + d_\gamma + D^-)} = 0. \tag{17}$$

Rearranging (17), we have the following quadratic equation for β:

$$\begin{aligned} & [(c_\beta + C^+ + d_\beta + D^+)(c_\gamma + C^- - c_\beta - d_\beta)]\beta^2 \\ + \; & [(c_\beta + C^+ + d_\beta)(d_\beta + d_\gamma + D^-) - (c_\gamma + C^- - c_\beta - d_\beta)(D^+ + d_\beta) + d_\beta D^+]\beta \\ - \; & d_\beta(d_\beta + D^+ + d_\gamma + D^-) \\ = \; & 0. \end{aligned} \tag{18}$$

Let

$$\begin{cases} a = (c_\beta + C^+ + d_\beta + D^+)(c_\gamma + C^- - c_\beta - d_\beta) \\ b = (c_\beta + C^+ + d_\beta)(d_\beta + d_\gamma + D^-) - (c_\gamma + C^- - c_\beta - d_\beta)(D^+ + d_\beta) + d_\beta D^+ \\ c = -d_\beta(d_\beta + D^+ + d_\gamma + D^-) \\ \Delta = b^2 - 4ac \end{cases},$$

(18) becomes

$$a\beta_i^2 + b\beta_i + c = 0. \tag{19}$$

According to Theorem 1 in Appendix A, the MAP estimates can be solved from (19) and (16). The **MAP** estimates are:

$$\hat{\gamma} = \frac{d_\gamma + D^-}{c_\gamma + C^- + d_\gamma + D^- + (d_\beta \frac{1-\hat{\beta}}{\hat{\beta}} - c_\beta)},$$
$$\hat{\beta} = \begin{cases} -\frac{c}{b}, & a = 0 \\ \frac{-b+\sqrt{\Delta}}{2a}, & a \neq 0 \end{cases}.$$

5 Handling iCLIP data

5.1 Generative model

We can model the generation of iCLIP data using the probabilistic model shown in Figure S1B. Let us define a crosslink site by (i, j, k), where i, j, k denote the chromosome, the 0-based genomic coordinate, and the strandness of this crosslink site. Note that j always represents coordinate in the forward strand. Then A is the vector of read generating probabilities for all possible crosslink sites in the genome. We have

$$A = \{\, \alpha_{ijk} \mid \sum_{i,j,k} \alpha_{ijk} = 1,\ (i, j, k) \text{ belongs to the genome} \,\}.$$

To generate an iCLIP read, we first sample the crosslink site I based on A. Once we have an $I = (i, j, k)$, we can generate the read sequence from the genome based according to read generating parameters $\mathscr{E}$. Note that if the crosslink site is in the forward strand, i.e. $k = +$, the read starts from position $(i, j+1, k)$ and extends toward the 3' end. If the crosslink site is in the reverse strand, i.e. $k = -$, the read starts from position $(i, j-1, k)$ and extends in the oppsite direction.

To make Figure S1B a fully generative model, $\mathscr{E}$ needs to include both the sequencing error model discussed before and parameters describing cDNA fragment length and read length. However, as we will show later, for the purpose of allocating multi-mapping reads, we only need the sequencing error parameters.

It is worth pointing out that Figure **S1B** is a simplification of toeprinting model Figure **S1A**. We have to make this simplication because iCLIP data contain less information that we can use to describe the data generating process.

5.2 Expectation-Maximization-Smoothing algorithm

Our goal is to estimate the expected read count at each crosslink site. Let $C = \{\, c_{ijk} \mid (i, j, k) \text{ belongs to the genome} \,\}$, where c_{ijk} represents the expected read count at site (i, j, k). We can estimate C from data using the EM algorithm:

E step Allocate each read's weight to its alignments.

Denote w_{nijk} as the expected fraction of read n assigned to crosslink site (i, j, k), we calculate it by

$$w_{nijk} = \begin{cases} \frac{\alpha^{\text{old}}_{ijk} \cdot P(r_n | i,j,k)}{\sum_{(i',j',k') \in \pi_n} \alpha^{\text{old}}_{i'j'k'} \cdot P(r_n | i',j',k')}, & (i,j,k) \in \pi_n \\ 0, & (i,j,k) \notin \pi_n \end{cases},$$

where π_n represents the set of crosslink sites read n aligns to.

M step Re-estimate A based on expected weights of alignments.

We first calculate expected count at each site c_{ijk} by

$$c_{ijk} = \sum_{n=1}^{N} w_{nijk}.$$

We then re-estimate A by

$$\alpha^{\text{new}}_{ijk} = \frac{c_{ijk}}{\sum_{i',j',k'} c_{i'j'k'}}.$$

Unfortunately, the above EM algorithm will not give us satisfying results because we have too many parameters compared to the data in hand. Thus, we add a smoothing step after M step:

S step If we denote the estimated A from M step as A^{int}, where "int" shorts for "intermediate", the smoothed A estimates are

$$\alpha^{\text{new}}_{ijk} = \frac{1}{2\mathtt{w}+1} \sum_{j'=j-\mathtt{w}}^{j+\mathtt{w}} \alpha^{\text{int}}_{ij'k}\ .$$

In the S step, we adopt a moving average smoother: $\alpha^{\text{new}}_{ijk}$ is estimated as the average of $\alpha^{\text{int}}_{ij'k}$ values within a window centered at (i, j, k). The window size is set to $2\mathtt{w}+1$, where $\mathtt{w}$, the half window size, is specified by users. With the S step, our algorithm becomes an instance of the Expectation-Maximization-Smoothing (EMS) algorithm proposed in 1990 (Silverman et al., 1990).

Note that our proposed algorithm is similar to the popular ChIP-Seq multi-mapping read allocator, CSEM (Chung et al., 2011) — we both use the EMS algorithm. However, our algorithm is different from CSEM in three aspects:

1. Unlike ChIP-Seq, our input data are strand-specific. Thus, we cannot pool reads aligned to same position but different strand together.

2. Because iCLIP data can measure RNA-protein interaction at single nucleotide resolution, we should use a much smaller window size.

3. Instead of treating all alignments of a read equal, we distinguish alignments of different qualities based on learned sequencing error parameters.

5.3 Implementation

In this subsection, we will discuss several important implementation details. Let us denote the iCLIP data as $\mathscr{D}$. According to the number of alignments each read has, we can partition $\mathscr{D}$ into 4 disjoint subsets: $\mathscr{D}_0$, $\mathscr{D}_{\text{unique}}$, $\mathscr{D}_{\text{multi}}$, and $\mathscr{D}_{\text{filt}}$. $\mathscr{D}_0$ contains all unalignable reads; $\mathscr{D}_{\text{unique}}$ contains all reads that align uniquely to the genome; $\mathscr{D}_{\text{multi}}$ contains all reads that have more than 1 but no more than 100 alignments; $\mathscr{D}_{\text{filt}}$ contains reads with more than 100 alignments. We decide to discard reads in $\mathscr{D}_{\text{filt}}$ because these reads align to too many places.

First, we notice that only reads in $\mathscr{D}_{\text{multi}}$ contain ambiguity and thus need to be involved in the EMS iterations. Run EMS algorithm only on $\mathscr{D}_{\text{multi}}$ will speed up our algorithm significantly.

Second, for the purpose of allocating multi-mapping reads, we do not need to model read length and fragment length distributions. This is because: a) All alignments of the same read share the same read length, thus the probabilities of generating the read length from different alignments will be canceled out in the E step. b) Current iCLIP protocol produces single-end reads. For single-end reads, we need to calculate the probability of generating a fragment no shorter than the read length for each alignment. Fortunately, these probabilities are approximately the same and thus can be canceled out in the E step. If we have paired-end reads, provided that we only keep alignments with fragment lengths in a reasonable range, the impact of fragment length distribution will be neglectable compared to the sequencing error model.

Thus, we can approximate $\mathscr{E}$ by

$$\mathscr{E} \approx \{w_p(r_b, s_b), w_q(r_b, s_b), P_{trans}(n|c), P_{init}(s), P_I(b)\}.$$

Note that the read sequence generation process is exactly the same as described in Section 2 except that we do not model "catastrophic" reads here. We estimate $\mathscr{E}$ from uniquely mapped reads, $\mathscr{D}_{\text{unique}}$.

Lastly, to further reduce the computational workload, we cluster the reads in $\mathscr{D}_{\text{multi}}$ according to the number of alignments, aligned locations and alignment qualities. Because reads in the same cluster will always have the same expected weights, we only need to calculate alignment weights once for each cluster. Alignment quality is obtained by binning the normalized read sequence generating probability. We partition $[0,1]$ into 10 equal sized bins with labels $0, 1, \ldots, 9$. For an alignment (i, j, k) of r_n, we find which bin the normalized probability $\frac{P(r_n|i,j,k)}{\sum_{(i',j',k')\in r_n} P(r_n|i',j',k')}$ falls in and that bin's label becomes the alignment's quality. In addition,we use the read sequencing generating probabilities, $\{P(r_n|i, j, k)\}$, of the first read in each cluster as each cluster's sequence generating probabilities.

In summary, Algorithm 3 describes the EMS algorithm we use. The default values for `ROUNDS` and `w` are

$$\texttt{ROUNDS} = 100, \quad \text{and} \quad \texttt{w} = 25.$$

Algorithm 3 The EMS algorithm

Input: iCLIP data $\mathscr{D}$, number of iterations ROUNDS, and half window size w.

Estimate sequencing error model $\mathscr{E}$ from $\mathscr{D}_{\text{unique}}$.

Set $\alpha_{ijk} = \frac{1}{2|G|}$, where $|G|$ is the size of the genome.

for ROUND $= 0$ **to** ROUNDS **do**

E step: For all $r_n \in \mathscr{D}_{\text{multi}}$ and $(i,j,k) \in \pi_n$, calculate

$$w_{nijk} = \frac{\alpha_{ijk} \cdot P(r_n|i,j,k)}{\sum_{(i',j',k') \in \pi_n} \alpha_{i'j'k'} \cdot P(r_n|i',j',k')}.$$

MS step: For all (i,j,k), calculate

$$c_{ijk} = \sum_{n=1}^{N} w_{nijk},$$

$$\alpha_{ijk} = \frac{1}{2\mathtt{w}+1} \cdot \frac{\sum_{j'=j-\mathtt{w}}^{j+\mathtt{w}} c_{ij'k}}{\sum_{i'j'k'} c_{i'j'k'}}.$$

end for

Output: $C = \{c_{ijk}\}$.

Appendix A Theorems

Theorem 1. *The MAP estimates exist and are the only solution of* (19) *and* (16)*. In addition, the MAP estimates are*

$$\hat{\gamma} = \frac{d_\gamma + D^-}{c_\gamma + C^- + d_\gamma + D^- + (d_\beta \frac{1-\hat{\beta}}{\hat{\beta}} - c_\beta)},$$

$$\hat{\beta} = \begin{cases} -\frac{c}{b}, & a = 0 \\ \frac{-b+\sqrt{\Delta}}{2a}, & a \neq 0 \end{cases},$$

Proof: We first show that the MAP estimates exist and are one set of the solutions for (19) and (16). Let us focus on the continuous function l_{MAP} at the closed domain $[\frac{1}{n}, 1-\frac{1}{n}]^2$. According to the Extream Value Theorem, l_{MAP} must have a maximum on $[\frac{1}{n}, 1-\frac{1}{n}]^2$. In addition, for large enough n, $\frac{\partial l_{\text{MAP}}}{\partial \gamma}|_{\gamma=\frac{1}{n}} > 0$, $\frac{\partial l_{\text{MAP}}}{\partial \beta}|_{\beta=\frac{1}{n}} > 0$, $\frac{\partial l_{\text{MAP}}}{\partial \gamma}|_{\gamma=1-\frac{1}{n}} < 0$, and $\frac{\partial l_{\text{MAP}}}{\partial \beta}|_{\beta=\frac{1}{n}} < 0$, thus we can conclude that the maximum does not locate at the boundary. Therefore, the maximum must be one of the stationary points. When we let $n \to \infty$, we have that the MAP estimates exist and are one set of solutions for (19) and (16).

We then show that (19) and (16) has only one set of solutions, which is mentioned in this Theorem. We show this case by case.

When $a = 0$, by rearranging (19), we obtain that $\hat{\beta} = -\frac{c}{b}$.

When $a \neq 0$, we must have $\Delta \geq 0$. Otherwise, we would not have any stationary points.

If $\Delta = 0$, $\hat{\beta} = -\frac{b}{2a} = \frac{-b+\sqrt{\Delta}}{2a}$.

If $\Delta > 0$, we may have two roots for (19). Let

$$\beta_1 = \frac{-b+\sqrt{\Delta}}{2a}, \qquad \beta_2 = \frac{-b-\sqrt{\Delta}}{2a},$$

be the two possible roots of (19). We need to further split $a \neq 0$ into two cases: $a > 0$ and $a < 0$.

1) $a > 0$. Note that $c < 0$ because d_β, d_γ are positive and D^+, D^- are non-negative. Thus, it holds that $\beta_2 < 0$ because

$$a>0, c<0 \Rightarrow b^2-4ac>b^2 \Rightarrow \sqrt{\Delta}>|b| \Rightarrow \beta_2=\frac{-b-\sqrt{\Delta}}{2a}<\frac{-b-|b|}{2a}\leq 0.$$

Therefore, $\hat{\beta}=\beta_1=\frac{-b+\sqrt{\Delta}}{2a}$.

2) $a<0$. It must hold that $b>0$ and $\sqrt{\Delta}<|b|$ because

$$a<0 \Rightarrow c_\gamma+C^- -c_\beta-d_\beta<0 \Rightarrow b>0,$$
$$a<0, c<0 \Rightarrow b^2-4ac<b^2 \Rightarrow \sqrt{\Delta}<|b|.$$

Therefore, we have $0<\beta_1<\beta_2$ because

$$b>0, \sqrt{\Delta}<|b| \Rightarrow -b\pm\sqrt{\Delta}<0 \overset{a<0}{\Longrightarrow} 0<\beta_1<\beta_2.$$

Without loss of generality, we suppose $0<\beta_1<\beta_2<1$. Otherwise, β_1 must be the only root of (19) and the proof is complete. Let us denote by γ_1 and γ_2 the γ values calculated from (16) with $\beta=\beta_1$ and $\beta=\beta_2$. Because

$$0<\beta_1<\beta_2<1 \Rightarrow \frac{1-\beta_1}{\beta_1}>\frac{1-\beta_2}{\beta_2} \Rightarrow d_\beta\frac{1-\beta_1}{\beta_1}-c_\beta>d_\beta\frac{1-\beta_2}{\beta_2}-c_\beta,$$

either (γ_1,β_1) is the only stationary point of (19) and (16) or $0<\gamma_1<\gamma_2<1$. We will show that the latter case is impossible by contradiction.

Supposing $0<\beta_1<\beta_2<1$ and $0<\gamma_1<\gamma_2<1$, l_{MAP} has two valid stationary points: (γ_1,β_1) and (γ_2,β_2). Let us first fix $\gamma=\gamma_1$ and increase β from β_1 to β_2. Because $\frac{\partial l_{\text{MAP}}}{\partial\gamma}=0$ at (γ_1,β_1) and increasing β decreases $\frac{D^+(1-\beta)}{\gamma+\beta-\gamma\beta}$, the partial derivative $\frac{\partial l_{\text{MAP}}}{\partial\gamma}<0$ at (γ_1,β_2). Then let us fix $\beta=\beta_2$ and increase γ from γ_1 to γ_2. Because increasing γ increases $\frac{c_\gamma+C^-}{1-\gamma}$ and $\frac{C^+}{1-\gamma}$, and decreases $\frac{d_\gamma+D^-}{\gamma}$ and $\frac{D^+(1-\beta)}{\gamma+\beta-\gamma\beta}$, we have $\frac{\partial l_{\text{MAP}}}{\partial\gamma}<0$ at (γ_2,β_2), which contradicts the assumption that (γ_2,β_2) is also a stationary point.

□