Enhancing transcriptome expression quantification through accurate assignment of long RNA sequencing reads with TranSigner

Hyun Joo Ji[1,2] and Mihaela Pertea[1,2,3]
[1]Center for Computational Biology, Johns Hopkins University; Baltimore, MD
[2]Department of Computer Science, Johns Hopkins University; Baltimore, MD
[3]Department of Biomedical Engineering, Johns Hopkins University; Baltimore, MD
Email: hji20@jh.edu, mpertea@jhu.edu

Running title: Accurate assignment of long RNA sequencing reads to transcripts

Abstract

Recently developed long-read RNA sequencing technologies promise to provide a more accurate and comprehensive view of transcriptomes compared to short-read sequencers, primarily due to their capability to achieve full-length sequencing of transcripts. However, realizing this potential requires computational tools tailored to process long reads, which exhibit a higher error rate than short reads. Existing methods for assembling and quantifying long-read data often disagree on expressed transcripts and their abundance levels, leading researchers to lack confidence in the transcriptomes produced using this data. One approach to address the uncertainties in transcriptome assembly and quantification is by assigning reads to transcripts, enabling a more detailed characterization of transcript support at the read level. Here, we introduce TranSigner, a versatile tool that assigns long reads to any input transcriptome. TranSigner consists of three consecutive modules performing: read alignment to the given transcripts, transcripts, computation of compatibility scores based on alignment scores and positions, and execution of an expectation-maximization algorithm to probabilistically assign read fractions to transcripts while estimating transcript abundances. Using simulated and experimental datasets from three well studied organisms — *Homo Sapiens*, *Arabidopsis thaliana* and *Mus musculus* — we demonstrate that TranSigner achieves higher accuracy in transcript abundance estimation and read assignment compared to existing tools.

Introduction

Long-read RNA sequencing (RNA-seq) represents a remarkable advancement towards achieving full-length sequencing of transcripts, offering novel insights into transcriptomes previously characterized using only short reads. Short-read data has limitations in several applications such as transcript assembly and quantification, primarily due to its fragmented nature and inherent biases (e.g., GC content, amplification) that add noises to downstream analyses (Benjamini & Speed, 2012; Hansen et al., 2010; Li et al., 2009). Long-read sequencing technologies address these limitations by substantially increasing the read lengths, allowing each read to generally cover a full-length transcript, and by employing strategies such as direct RNA sequencing to reduce biases. Consequently, long-read data can provide more comprehensive and accurate profiles of complex transcriptomes.

However, despite their potential, the full capabilities of long-read RNA-seq remain untapped due to the limited inventory of tools optimized for analyzing long-read evidence. Although tools such as FLAIR (Tang et al., 2020), Bambu (Chen et al., 2023), ESPRESSO (Gao et al., 2023), and StringTie (Kovaka et al., 2019) are designed to characterize transcriptomes by both identifying novel isoforms as well as quantifying transcripts using long-read RNA-seq data, their results lack agreement (Chen et al., 2023; Gao et al., 2023; Pardo-Palacios et al., 2023; Tang et al., 2020). These discrepancies often arise from false positive transcripts, preventing users from being confident in their predicted set of transcripts.

One way to address uncertainties in transcriptome assemblies is by assigning specific long reads to transcripts. This allows for a more in-depth evaluation on the read-level support for transcripts, as opposed to relying on read counts only. Given read-to-transcript assignments, transcripts can be associated with a distribution of supporting read lengths, quality scores, alignment positions, and more. These expanded sets of features can be used to derive a more confident set of transcripts and improve the accuracy of transcript abundance estimates.

Few tools, including FLAIR and Bambu, track read-to-transcript assignments, but this functionality is integrated into more complex pipelines that also identify novel isoforms in addition to quantifying known transcripts. A standalone tool capable of performing quantification and read-to-transcript assignments on any input transcriptome can be paired with other methods focusing on transcriptome assembly, enabling

users to investigate any transcriptome of their choice. However, this need is largely unmet, with only one recent method – NanoCount (Gleeson et al., 2021) – attempting to address it.

Here we introduce TranSigner, a novel method for accurately assigning long RNA-seq reads to a given transcriptome. TranSigner first maps reads onto the transcriptome using minimap2 (Li, 2018, 2021) and extracts features from the alignments, including alignment scores and the 3' and 5' end read positions on a transcript. These features are then utilized to compute compatibility scores for read and transcript pairs, indicating the likelihood of a read originating from a specific transcript. TranSigner then employs an expectation-maximization (EM) algorithm to derive maximum likelihood (ML) estimates for both the read-to-transcript assignments and transcript abundances simultaneously. We show that by guiding the EM algorithm in the expectation step with precomputed compatibility scores, TranSigner generates high-confidence read-to-transcript mappings and improves transcript abundance estimates.

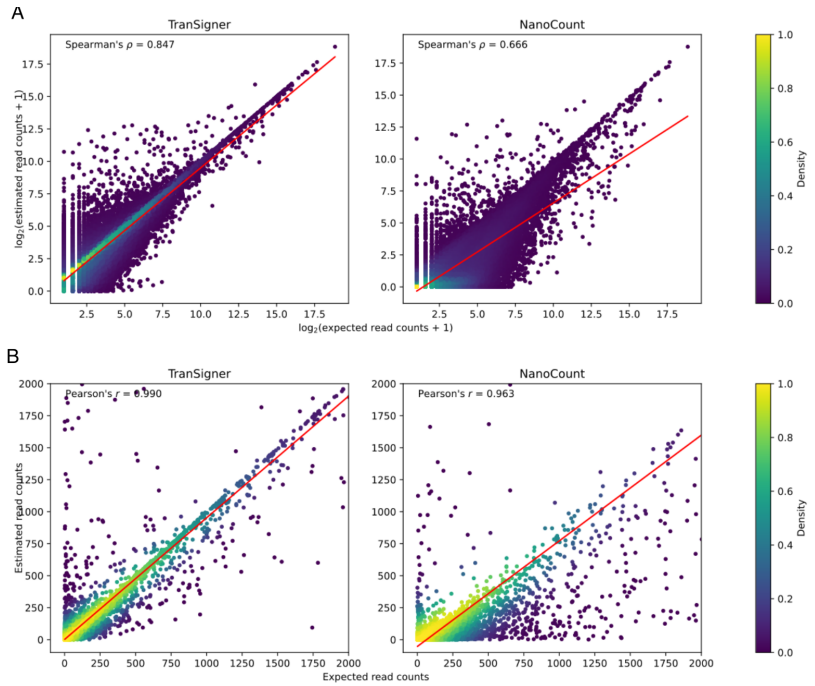

Figure 1. Correlation scatter plots comparing read count estimates generated by TranSigner and NanoCount's on one simulated direct RNA read set. Both tools were provided with the full RefSeq annotation from which the reads were generated from. A: Scatter plots showing the nonlinear correlations between the log transformed ground truth and estimated read counts. Spearman's correlation coefficient values are shown at the top of each plot. B: Scatter plots showing the linear correlations between ground truth and estimated read counts. x- and y- axes were limited to [0, 2000] range for demonstration. Pearson's correlation coefficient values are shown at the top of each plot.

Results

Simulated data performance. TranSigner is a transcript quantification-only tool that requires an input transcriptome and a long-read RNA-seq sample to evaluate the levels of the expressed transcripts. We first compared TranSigner against another quantification-only tool, NanoCount (Gleeson et al., 2021), which was designed to improve the expression quantification from nanopore (ONT) direct RNA sequencing. Consequently, we benchmarked both these tools using three independently simulated sets of ONT direct RNA reads. We simulated reads from transcripts in the RefSeq annotation (release 110), and then provided the full RefSeq annotation as the target transcriptome to both TranSigner and NanoCount (see Methods for a full description of the simulated datasets). From now on, we'll refer to the transcripts from which the reads were simulated as the origin transcripts.

We conducted both linear and nonlinear correlation analyses comparing the expected read counts to each tool's estimates. Linear correlation analysis, utilizing Pearson's correlation coefficient (PCC), evaluates the ability to assign reads equally, regardless of transcript abundance, while nonlinear correlation analysis, employing Spearman's correlation coefficient (SCC) on log-transformed read counts, assesses how well estimates capture monotonic trends in gene expression patterns. In both analyses, we observed that TranSigner's estimates had stronger correlations with the ground truth compared to NanoCount's, as illustrated in Figure 1, which shows results from one dataset typical of all three simulated datasets (see Supplementary Tables S1 for SCC and PCC values across all three simulated read sets). In both raw (Figure 1A) and log-transformed (Figure 1B) read count correlation scatter plots, TranSigner showed higher concentrations of dots near the diagonal. However, this feature is not observed in the plots of NanoCount's results, which generally show a tendency for NanoCount to underestimate the read counts. On the simulated direct RNA datasets, TranSigner's average SCC and PCC values were 0.847 and 0.99, whereas NanoCount's were 0.665 and 0.963 (Supplementary Table S1).

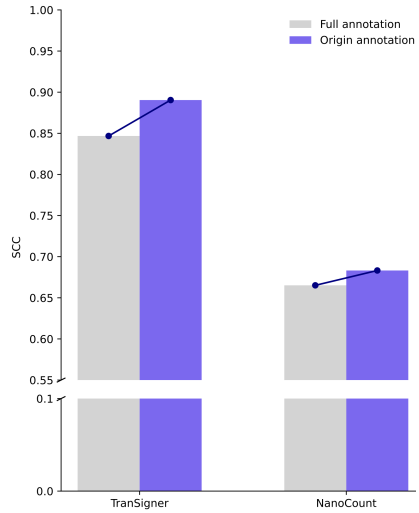

Figure 2. Spearman's correlation coefficient (SCC) values observed when either the origin transcriptome (in blue) or the full RefSeq annotation (in grey) is used to run TranSigner and NanoCount on simulated direct RNA reads. Average SCC values across three read sets are shown.

Even for extensively studied species, gene annotation catalogs are often incomplete, missing both potential gene loci and many transcript isoforms (Amaral et al., 2023; Varabyou et al., 2023). This is one reason why, unlike TranSigner and NanoCount, most existing tools for quantifying transcripts with long-read RNA-seq data prioritize identifying novel isoforms first. This approach can theoretically lead to better quantification of expressed transcripts, as demonstrated by our results in Figure 2, where we show that the correlation coefficients between estimated and true read counts generally improve for both TranSigner and NanoCount when the origin transcripts are provided in the input instead of the full reference annotation. However, achieving an accurate transcriptome remains a challenging problem, with different tools obtaining varying accuracies in this task, while also relying to varying degrees on the input reference annotation. Genome-guided transcriptome assemblers like StringTie (Kovaka et al., 2019) can

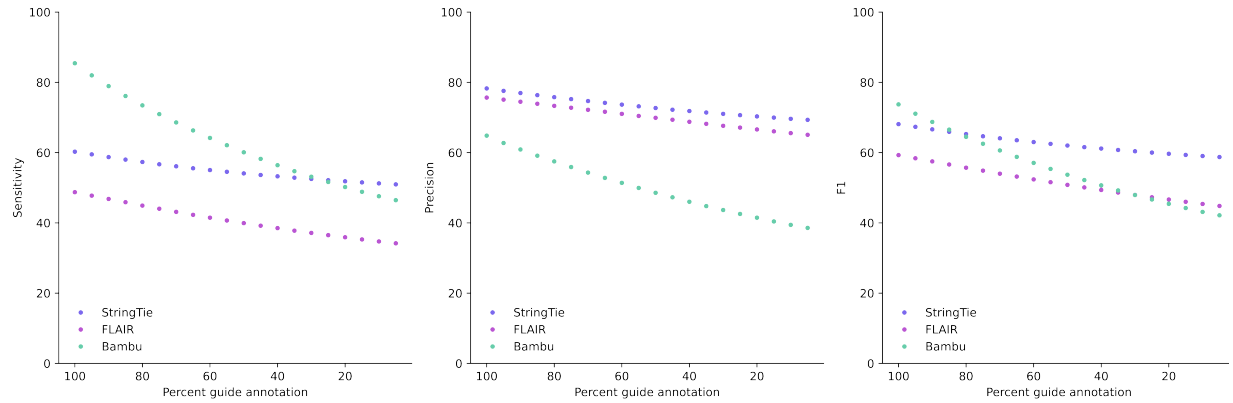

Figure 3. Long-read assembly accuracies of StringTie, FLAIR, and Bambu with varying percentages (5% to 100%) of randomly sampled origin transcripts provided in the input guide annotation. For each set of simulated ONT direct RNA reads, guide annotations were sampled three times, yielding 9 (3 datasets of reads with 3 guide samplings each) independent observations for each percentage. Mean values across these 9 observations are shown as circles for all three metrics – sensitivity, precision, and F1 – from left to right.

reliably profile a transcriptome in the absence of an input guide annotation, while methods like Bambu (Chen et al., 2023) or FLAIR (Tang et al., 2020) demonstrate a substantial decrease in both sensitivity and precision of transcript identification when the percentage of origin transcripts in the input guide annotation is reduced (see Figure 3 and Methods). While Bambu may appear to outperform other tools when the transcriptome closely matches the reference annotation, this assumption would most likely not hold true for studies involving poorly annotated organisms or in cases where the RNA-seq sample contains many novel isoforms. StringTie, on the other hand, consistently outperforms other tools across varying levels of origin transcripts present in the input annotation. However, StringTie does not assign specific long reads to the transcripts that it assembles. Our aim in designing TranSigner was to provide both the capacity to assign long-reads to individual transcripts and to improve the quantification accuracy of the assembled transcriptome.

To assess TranSigner's performance in quantifying an input transcriptome, we compared its abundance estimates when applied directly to a guide annotation reference or to the transcriptome assembled by StringTie (denoted as StringTie + TranSigner), with those obtained by NanoCount, Bambu, and FLAIR using the same guide reference. The StringTie + TranSigner combination was introduced to compare TranSigner's performance against other tools, such as FLAIR or Bambu, capable of novel isoform identification. We benchmarked all tools using the same three sets of ONT direct RNA simulated reads that were used to evaluate TranSigner and NanoCount. For every read set, we ran all tools 60 times, each time with a different guide annotation obtained by randomly sampling the RefSeq annotation to include varying percentages of the original transcripts. We selected 20 different percentages between 5% and 100%, with increments of 5%. For each percentage, we independently sampled the annotation three times. Correlation coefficients between the true and estimated read counts are shown in Figure 4. Except for StringTie + TranSigner, every tool experienced a drastic drop in SCC values as the percentage of origin transcripts decreases. When 90% or fewer origin transcripts were provided, StringTie + TranSigner yielded the best SCC values (Figure 4A), demonstrating that this combination is the best in preserving the rank of the expression values. StringTie doesn't output read counts for its transcript abundances estimates, so it was excluded from this nonlinear correlation analysis. However, as illustrated by Figure 4B, TranSigner obtains better read per base coverage estimates when run on the StringTie-assembled transcriptome. Although TranSigner doesn't include these coverages in its outputs, we post-processed the read-to-transcript assignments to obtain these estimates (see Methods). Specifically, its re-estimates exhibit stronger linear correlations with the ground truth compared to the initial values computed by StringTie.

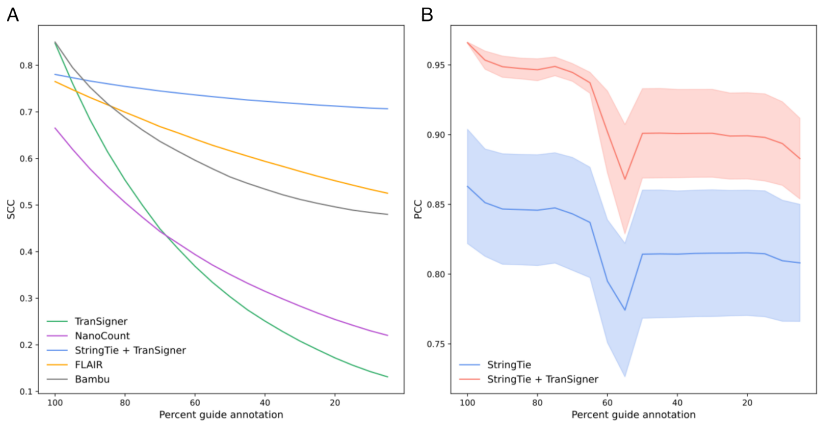

Figure 4. Correlation analyses between true and estimated abundances, conducted on three simulated ONT direct RNA read sets. A: SCC values quantifying the nonlinear correlation between the ground truth and the read count estimates outputted by TranSigner, NanoCount, StringTie + TranSigner, FLAIR, and Bambu across varying percent guide annotations. B: PCC values quantifying the linear correlation between the ground truth and the read coverage estimates outputted by StringTie and StringTie + TranSigner, across varying percent guide annotations. Lines represent the average correlation coefficient values (SCC in A and PCC in B) across 9 independent observations (3 read sets, 3 guide samplings). Shaded areas in B represent the standard error of the mean interval for average PCC values.

One key feature of TranSigner is its ability to assign specific reads to transcripts, particularly useful in experiments where users need to identify reads originating from specific transcripts of interest. In this context, we compared TranSigner and StringTie + TranSigner with FLAIR and Bambu, which also output read-to-transcript assignments. Their performance was evaluated using recall, precision, and F1 scores, computed by counting the number of correctly versus incorrectly assigned number of reads (see Methods). When guide annotations retained over 90% of the ground truth, TranSigner had the highest recall (Figure 5). However, as guides became increasingly incomplete, StringTie + TranSigner consistently showed higher recall values than other methods. Bambu outperformed the rest in terms of precision by a small margin, although the standard error of the mean (SEM) interval of StringTie + TranSigner eventually matched Bambu's (Figure 5B). Overall, TranSigner achieved the highest F1 scores with nearly complete guides, but its performance declined rapidly as the number of origin transcripts in the guides decreased, as expected. StringTie + TranSigner showed the second-best F1 scores and proved resilience to incomplete guides, making it the preferred choice when approximately 90% or fewer of the origin transcripts were available as guides (Figure 5C).

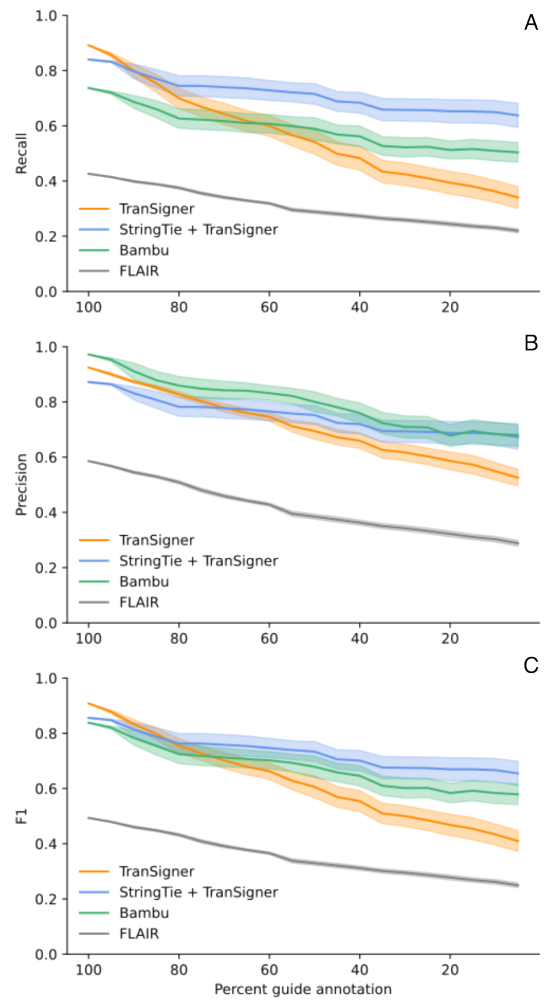

Figure 5. Average recall (A), precision (B), and F1 (C) values computed for the read-to-transcript assignments generated by TranSigner, StringTie + TranSigner, Bambu, and FLAIR on ONT direct RNA simulated data sets. SEM intervals are shown as shaded areas.

Real data performance. We further evaluated the performance of TranSigner and StringTie + TranSigner using experimental data. Specifically, we utilized a dataset provided by the Singapore Nanopore Expression Project (SG-Nex), consisting of ONT direct RNA reads sequenced from the K-562 lymphoblast cell line (Chen et al., 2021). This dataset includes synthetic spike-in transcripts, known as sequins, with known annotation and concentrations. As ground truth for this experiment, we used the CPM values provided by SG-Nex and compared them with the estimates obtained by TranSigner, StringTie + TranSigner, and Bambu. We ran each tool twice by using two different input guides: one including the full sequin annotation in addition to the GRCh38 reference annotation and the other containing only the GRCh38 reference transcripts without the sequins. The guide annotation without the sequins reflects real-world scenarios where transcript annotations are absent from the reference.

In nonlinear correlation analyses, TranSigner achieved an SCC of 0.75, surpassing both Bambu (0.72) and StringTie + TranSigner (0.62) when provided with the full sequin annotation. However, when no sequin annotation was provided, StringTie + TranSigner outperformed Bambu, obtaining an SCC value of 0.54, compared to Bambu's SCC value of 0.43. This trend persisted in linear correlation analyses, with TranSigner achieving the highest PCC value with full annotation, while StringTie + TranSigner's was the best performer in the absence of sequin annotation (Figure 6A). Overall, these results suggest that StringTie + TranSigner may be preferable in scenarios

where numerous unannotated or novel isoforms are anticipated, while TranSigner is optimal when the reference is presumed to be nearly complete.

Finally, we expanded our evaluation to include publicly available paired short and long read datasets from two well-studied species: *Arabidopsis thaliana* and *Mus musculus*. Each pair consists of a short-read dataset and a long-read dataset from the same sample. The short-read libraries were all generated through poly-A selection and sequenced with Illumina sequencers while the long reads were generated using ONT direct RNA or cDNA sequencing protocols. Unlike the sequin samples or simulated long reads, the ground truth is unknown for these datasets as we lack information about the expressed transcripts and their concentrations. However, it is generally assumed that short reads provide more accurate abundance estimates compared to long reads, as they are less error-prone, and they typically yield more reads. Consequently, we assessed the long read-based abundance estimates by comparing them to the short read-based abundance estimates and assumed that a higher correlation between long read- and short read-based abundance estimates is indicative of a more accurate long-read abundance estimate.

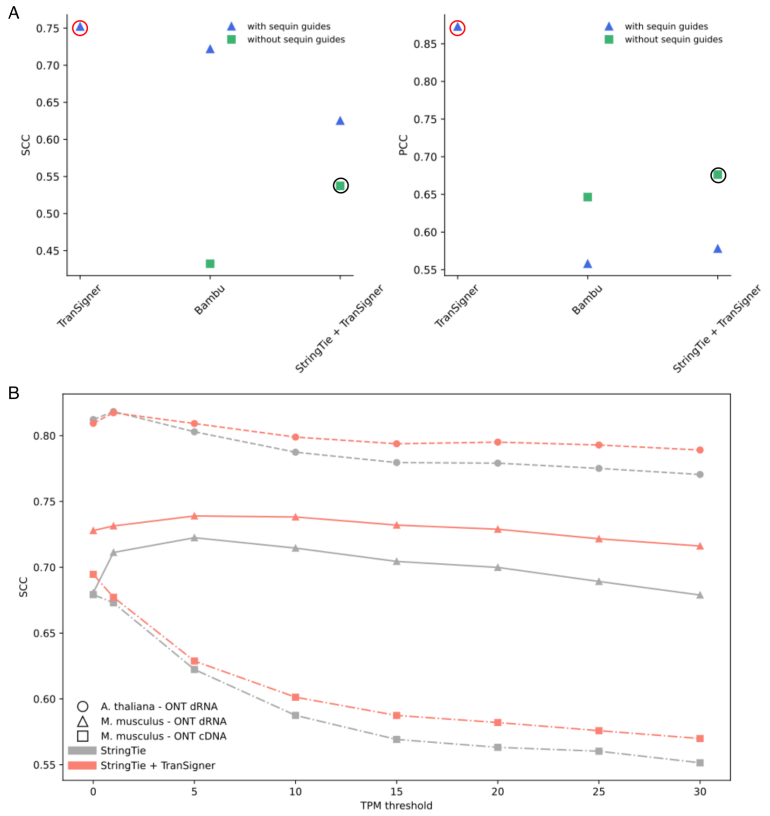

Figure 6. TranSigner's performance on two types of experimental datasets: sequin data and short- and long-read paired data set. A: Spearman (SCCs) and Pearson correlation coefficients (PCCs) between expected and estimated CPM values computed by TranSigner, Bambu, and StringTie + TranSigner on sequin data. Blue triangles indicate the correlation coefficient given the full sequin annotation, while green squares indicate results obtained when no sequin annotation was provided. Red and black circles show the highest coefficient with or without the sequin annotation, respectively. B: SCCs computed between short- and long-read-based TPM estimates on pairs of Illumina short- and ONT long-read samples from *A. thaliana* and *M. musculus* across multiple TPM thresholds on the short-read quantified RefSeq transcripts. Average values across 3 pairs of short- and ONT dRNA long-read samples are shown for *A. thaliana*, and average across 4 pairs of short- and ONT dRNA and 3 pairs of short- and ONT cDNA long-read samples are separately shown for *M. musculus*.

We conducted nonlinear correlation analyses between the short read-derived TPM estimates and long-read derived TPM estimates. Unlike StringTie, TranSigner doesn't include TPMs in its output, so we processed TranSigner's read counts to obtain counts per million (CPM) estimates, which are equivalent to TPMs in a long-read RNA-seq experiment where each read is considered to represent a transcript (see Methods). We used salmon (Patro et al., 2017) to obtain TPM estimates on the transcriptome defined by the RefSeq release 110 annotation on GRCh38, using the Illumina short-read datasets.

As expected, using TranSigner to re-estimate the TPM of transcripts assembled by StringTie resulted in better correlation with short-read TPM estimates compared to StringTie's initial estimates. Since we don't have definitive knowledge of which transcripts are present in each dataset, we focused our comparison on transcripts that were assembled in both short- and long-read samples, as these were more likely to be correctly assembled. Additionally, as transcripts with low abundances are prone to misassembly with short-read data and are often excluded from downstream analyses, we evaluated the performance of StringTie and StringTie + TranSigner's quantification at progressively increasing TPM thresholds: 1, 5, 10, 15, 20, 25, and 30. As illustrated in Figure 6B, when TranSigner was applied to StringTie's output, it achieved higher nonlinear correlations between short- and long-read TPM estimates, with the increase in SCC values being more pronounced for higher TPM thresholds.

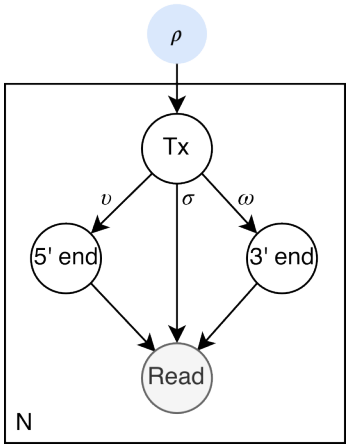

Figure 7. Graphical representation of TranSigner's long-read RNA-seq model. Empty circles denote latent variables, the shaded circle represents the observed variable, and the blue circle indicates the primary parameter of the model – specifically, the relative abundance of the transcript. Parameters υ, ω approximate the likelihood of the specific 5' and 3' end positions of the read on the transcript, while parameter σ models the likelihood of observing a specific read sequence given a transcript and the read's end positions. N represents the total number of reads generated in a single long-read RNA-seq experiment.

Discussion

Assigning long reads to transcripts is a challenging task that involves effective resolution of multi-mapping reads. Recent studies have unveiled the growing complexity of eukaryotic transcriptomes, revealing numerous isoforms across gene loci. The introduction of long-read RNA-seq technologies promises to uncover even more novel isoforms, as reads produced by these methodologies can capture full-length transcripts, overcoming the limitations of short reads. Although long reads cover transcripts at greater lengths, technical artifacts such as base calling errors and end truncations prevent these reads from being accurately mapped to their origins. With TranSigner, we've developed several strategies to address this challenge, facilitating the correct assignment of reads that ambiguously map to multiple isoforms.

Additionally, we designed TranSigner to complement another method capable of transcriptome assembly. As gene annotation is still an unresolved issue, determining the accuracy and completeness of a profiled transcriptome remains difficult. Users often struggle to select the appropriate reference for their analyses, leading to unpredictable impacts on their results. In our study, we observed that assembly qualities can drop drastically when less complete guides are provided, indicating that some tools overly dependent on high-quality reference annotations may not perform well in real-world scenarios where many novel isoforms are expected. By introducing a standalone tool for read-to-transcript assignments, we made these assignments easier to obtain regardless of the input transcriptome. Integrating this step into long-read RNA-seq data processing pipelines will improve the accuracy of transcriptomes identified using long reads by allowing users to inspect the quality of the reads supporting the transcripts and filter out less-supported transcripts. This, in turn, will lead to more accurate abundance estimates, as our results demonstrate the significant influence of assembly accuracy on correctly identifying transcript abundances.

Methods

Long-read RNA-seq model. We describe the long-read RNA-seq process using a generative model (Figure 7). The conceptualization of RNA-seq as a generative process in which reads are sampled from a

pool of transcripts has already been used in models for short-read quantification. We adopted the general framework proposed by others (Li et al., 2009; Pachter, 2011) but introduced necessary modifications to tailor the model to long read data.

Given a read, we assume that there are three unobserved events in the RNA-seq experiment that determines that a particular read's sequence: (1) the transcript from which that read was sequenced, (2) the position within the transcript of the 3' end of the read, and (3) the transcript position of the reads' 5' end. Our model, thus, associates each of observed reads with three latent variables: the transcript (T) from which the read was generated, its 3' end position (S) and 5' end position (E) in T. We want to estimate the parameter ρ, which represents the relative abundance of the transcript.

Given a set of transcripts $T = \{t\}$ where $|T| = M$, the complete data likelihood function of this model is:

$$\mathcal{L}(\rho) = \prod_{r \in R} \sum_{t \in T} \mathrm{P}(O_r = t|\rho)\mathrm{P}(s_{rt}|O_r = t)\mathrm{P}(e_{rt}|O_r = t)\mathrm{P}(r|O_r = t, s_{rt}, e_{rt})$$

where ρ is the set of relative transcript abundances defined as $\rho = \{\rho_t\}_{t \in T}$ where $\sum_{t \in T} \rho_t = 1$, R is the set of mapped reads defined as $R = \{r\}$ with cardinality of N, s_{rt} and e_{rt} are the 3' and 5' end positions of a read r in a transcript t, and O_r is a random variable denoting the origin transcript for read r. $\mathrm{P}(O_r = t|\rho) = \rho_t$, since in an RNA-seq experiment the probability to select a transcript t to sequence depends on its relative abundance. We'll approximate the 5' end and 3' end positions of a read in a transcript as the positions where the read alignment starts and ends on that transcript, respectively. Different long-read RNA-seq technologies show various biases towards the ends of the transcripts (Amarasinghe et al., 2020; Chen et al., 2021; Grünberger et al., 2022; Wongsurawat et al., 2022). Nonetheless, long reads are more likely to cover all bases of a transcript, compared to the short reads, which are generated from fragments of the transcript. The likelihood for a read's end position should decrease as its distance from the transcript end increases. We model this expectation using two position bias parameters – υ and ω for the 3' and 5' ends, respectively – to control how far apart the ends of a read can be from the ends of a transcript. For an alignment between a read r and a transcript t, we'll refer to the distances between the alignment ends and transcript ends as 'end distances' and denote them as δ_s^{rt} and δ_e^{rt} for the 5' and 3' ends, respectively. Then we define υ and ω as:

$$\begin{aligned} \mathrm{P}(s_{rt} = i|r \in t) \approx \upsilon_{rt} = 1 \text{ if } \left|\delta_s^{rt'} - \delta_s^{rt}\right| \leq \beta_s, 0 \text{ o.w.} \\ \mathrm{P}(e_{rt} = j|r \in t) \approx \omega_{rt} = 1 \text{ if } \left|\delta_e^{rt'} - \delta_e^{rt}\right| \leq \beta_e, 0 \text{ o.w.} \end{aligned} \tag{1}$$

where $\delta_s^{rt'}$ and $\delta_e^{rt'}$ represent the end distances for the primary alignment of read r and transcript t'. Here, t' represents the transcript to which read r aligns on the primary alignment, which might not be the same as transcript t. Since alignment positions are indexed from the 5' to 3' direction on transcript t, end distances are computed as $\delta_s^{rt} = s_{rt} = i$ and $\delta_e^{rt} = |t| - e_{rt} = |t| - j$ where $|t|$ is the length of transcript t. Parameter β represents the tolerance threshold on how much greater the end distances can be compared to the primary alignment's end distances for a given read r. This relative thresholding on end distances (δ) ensures that each read is compatible with at least one transcript after this filtering step since the primary alignment will always be considered "good," which wouldn't true if a constant threshold was uniformly applied for all reads. A bias term of 0 – either υ or ω – indicates that the corresponding (r, t) pair will be considered entirely incompatible, filtering it out from any downstream analysis. Moreover, the parameters for the 3' end are treated separately from those for the 5' end because sequencing behaves differently at these ends. For example, there is a stronger coverage bias towards the 3' end when

nanopore-based direct RNA sequencing protocols are employed (Amarasinghe et al., 2020; Chen et al., 2021; Grünberger et al., 2022; Wongsurawat et al., 2022).

The probability of observing a read r given all the latent variables is modeled using the alignment score between read r and transcript t (denoted by x_{rt}) as:

$$\mathrm{P}(r|r \in t, s_{rt} = i, e_{rt} = j) \approx \sigma_{rt} = \frac{x_{rt}}{\max_{k \in T} x_{rk}} \tag{2}$$

Note that if multiple alignments exist between read r and transcript t, we only retain the alignment with the maximum score. Using the above definitions, we can rewrite the likelihood function as:

$$\mathcal{L}(\rho) = \prod_{r \in R} \sum_{t \in T_r} \rho_t \upsilon_{rt} \omega_{rt} \sigma_{rt} \tag{3}$$

where T_r is the set of transcripts aligned to read r, with υ_{rt}, ω_{rt}, and σ_{rt} set to zero for any unaligned pair of read r and transcript t.

Alignment. We used minimap2 with parameter -N 181 to align the long reads to the set of input transcripts. The -N parameter was set to retain all primary and secondary alignments reported by minimap2. By default, minimap2, limits the maximum number of secondary alignments to 5. We observed that the number of true positives (correct read to transcript alignments) increases when this limit is increased so we set it to 181, the highest number of transcripts in a single gene locus according to the RefSeq release 110 annotation on the human GRCh38 genome. This strategy provides rough, preliminary estimates on the compatibility between reads and transcripts, without excluding any read and transcript pair for having suboptimal alignment scores. The user can freely adjust this parameter by specifying it in TranSigner's input, which will then pass it to minimap2.

We offer an additional approximate alignment option using a tool called PSA_aligner, first introduced as part of the MaSURCA genome assembler (Zimin et al., 2013). PSA_aligner returns the start and end coordinates of the longest common subsequence (LCS) between the query and the target, obtained by matching *k*-mers stored in a partial suffix array (PSA). An alignment is defined as the linear least squares fit through the matching *k*-mers. When this additional alignment is performed, minimap2 results are combined with PSA_aligner results.

PSA_aligner results lack alignment scores. Instead for each alignment a, we compute our own alignment score based on the query-to-target cover ratio (κ_a), defined as follows:

$$\kappa_a = {L_a^q}/{L_a^t}$$

where $L_a^i = S_a^i - E_a^i, S_a^i,$ and $E_a^i, i \in \{q, t\}$ represent the cover length, alignment start, and alignment end, respectively, of the query (q) and target (t). PSA_aligner outputs exactly one alignment between any read and transcript pair, so, for clarity, we'll denote κ_a by κ_{rt} for each alignment between a read *r* and transcript *t*.

We expect the ratio between the query and target cover lengths to be close to 1 if the LCS contains few or no gaps. We evaluate the alignment's quality by first computing how much κ deviates from the desired cover ratio of 1 and multiplying it by -1. Next, we normalize the query cover ratio losses of all alignments

for a given read r by the maximum (i.e., least negative) loss value observed for r, defining a score metric for the alignment found by the PSA_aligner as follows:

$$\sigma_{rt}^{PSA} = \frac{-|\kappa_{rt} - 1| + c_r}{\max_{j \in T_r} -|\kappa_{rj} - 1| + c_r}$$

where c_r is a positive constant padding defined as $1 + \min_{i \in T_r} -|\kappa_{ri} - 1|$ to ensure neither the numerator nor the denominator of this fraction is 0. This definition for σ^{PSA} enforces that the more deviation between the query and target cover lengths, the lower the σ^{PSA} value. We combine the PSA_aligner and minimap2 scores using a ratio π such that:

$$\sigma_{rt} = \pi * \sigma_{rt}^{MM2} + (1 - \pi) * \sigma_{rt}^{PSA} \tag{4}$$

where σ_{rt}^{MM2} follows the definition specified in equation (2). By default, π = 0.99, putting more weight on the minimap2 alignment results.

Alignment-guided expectation-maximization algorithm (AG-EM). Our primary goal is to accurately assign the long reads to their respective transcript origins. We represent the read-to-transcript assignments by the hidden variables Z_{rt} where $Z_{rt} = 1$ if $O_r = t$, $s_{rt} = i$, $e_{rt} = j$ and 0 otherwise. To estimate the values of the Z_{rt} variables we employ an expectation-maximization (EM) algorithm that computes the maximum likelihood (ML) estimates for the transcript abundances $\rho = \{\rho_t\}$ as follows.

Update rules. The EM algorithm consists of alternating expectation (E) and maximization (M) steps, repeated until convergence. During the E step, the expected values for Z_{rt}–at some iteration n–are computed as follows:

$$E_{Z|r,\rho^{(n)}}[Z_{rt}] = \alpha_{rt}^{(n)} = \frac{\rho_t^{(n)} \upsilon_{rt} \omega_{rt} \sigma_{rt}}{\sum_{t' \in T_r} \rho_{t'}^{(n)} \upsilon_{rt'} \omega_{rt'} \sigma_{rt'}} \tag{5}$$

where $\alpha = \{\alpha_{rt}\}_{r,t \in A}$ denotes the set of expected values for Z_{rt} and A is the set of alignments between all reads and transcripts. α has the property $\sum_{t \in T_r} \alpha_{rt} = 1$, which allows us to interpret the expected values as fractions of reads assigned to the transcripts they are compatible with. In the following M step, then, the fragments of reads assigned to each transcript is summed up and then normalized by the total number of transcripts to get the relative transcript abundances, expressed as:

$$\rho_t = \frac{\sum_{r \in R_t} \alpha_{rt}}{\sum_{r',t' \in A} \alpha_{r't'}} \tag{6}$$

where R_t is the set of reads aligned to transcript t. The denominator is constant across iterations and is equivalent to the total number of reads in a long-read RNA-seq experiment where each read represents a transcript, so we precompute this value prior to EM.

Initialization. Prior to the EM iterations, the relative transcript abundances (ρ) are initialized to the uniform distribution:

$$\rho_t = \frac{1}{|T_A|}$$

where T_A is the set of transcripts with at least one alignment to a read in R. Additionally, the values for υ, ω, and σ don't change during iterations, so we precompute their values and store them separately in a matrix X of dimensions N rows and M columns. For simplicity, we'll refer to X as the compatibility score matrix. The computation specified in equation (5) is further simplified as:

$$\alpha_{rt}^{(n)} = \frac{\rho_t^{(n)} X_{rt}}{\sum_{t' \in T_r} \rho_{t'}^{(n)} X_{rt'}} \tag{7}$$

The precomputation step involves a single scan over the alignment results, extracting values such as the alignment scores and alignment start / end positions, and then applying the definitions provided in equations (1), (2), and, optionally, (4) if the PSA_aligner is employed.

Optimization. Once X is precomputed and ρ is initialized, EM iterations are repeated until convergence, i.e., until the total sum of changes in the relative transcript abundances is less than a predefined threshold, by default set at 0.005. The user can adjust this threshold to increase the accuracy of the ML estimates at the expense of speed.

The novelty of our method comes from guiding the EM algorithm with the priors extracted from the alignment results, as detailed in the E-step update rule shown in equation (5). To further amplify the impact of these priors, we implemented an algorithm called drop. The drop algorithm (see Supplementary Figure S1) sets $X_{rt} = 0$ if the fraction of read r that is assigned to transcript t (i.e., α_{rt}) gets below a threshold, $\tau \in [0,1]$. This effectively drops the compatibility relationship between read r and transcript t and ensures that no fraction of r gets assigned to t in any iterations following the drop, as α_{rt} will always be 0 since its computation involves multiplication by X_{rt} (equation 7). After the drop, another E-step is performed with the updated X scores to recompute the new α_{rt} values. The τ value depends on the read r considered, and by default:

$$\tau_r = \frac{1}{|T_r|}$$

where T_r is the set of transcripts that are compatible with r. The drop algorithm is called only right after the first E-step calculation, and its purpose is to discard minimap2 alignments that are not robust. The drop algorithm offers the potential to achieve a higher optimum compared to a naïve EM algorithm (Pachter, 2011), which relies solely on the relative transcript abundances (ρ) in its E step update.

Read assignment. We can use the α values estimated by the EM algorithm to infer read assignments to transcripts. Raw α values represent fractional read assignments, where a single read may be distributed among multiple transcripts. These assignments might be challenging to interpret, as we assume each read to originate from a single transcript. To increase the interpretability and usability of the α values, we implemented the push algorithm (see Supplementary Figure S2). This algorithm processes raw α values, converting them into hard assignments where each read is assigned to exactly one transcript. The push algorithm iterates through the reads and pairs each of them to the transcript with the highest read fraction as shown by the corresponding α value. It then recomputes the relative transcript abundances based on these hard assignments. These new α and ρ values may deviate from their EM-derived ML estimates, potentially resulting in reduced accuracy. We tested this using simulated data and observed only negligible reductions in accuracy.

Implementation. TranSigner requires two inputs: a GTF file containing a reference gene annotation of the target transcriptome and a FASTQ file containing long RNA-seq reads. The reference annotation can be obtained from public sources such as RefSeq (O'Leary et al., 2016), GENCODE (Frankish et al., 2019), or CHESS (Varabyou et al., 2023), or it can be derived from transcriptome assemblies produced by programs like StringTie. The latter annotations have the advantage of including novel isoforms while restricting the annotated transcripts to only those found to be expressed in the analyzed sample.

As illustrated in Figure 9, TranSigner consists of three modules: align, prefilter, and em. In the align module, input long reads are aligned to the target transcriptome using minimap2. The resulting alignment file becomes the input for the next module. Next, in the prefilter module, TranSigner extracts features such as the 3' and 5' end alignment positions and the ms alignment scores computed by minimap2. These features are used to compute the initial values for the fractions of reads assigned to transcripts (α), a compatibility matrix between transcripts and reads, as well as an index of the IDs of the transcripts found to be compatible with reads in the align module, which represent a subset of the target transcriptome.

Optionally, the PSA_aligner can be run in addition to minimap2 in the prefilter module before α is initialized. When the PSA_aligner is run, users can set the ratio that defines how much impact each aligner result has on α. By default, we set this ratio to 0.99:0.01 (minimap2 : PSA_aligner). In this case, the PSA_aligner results and the cover ratios (κ) are also outputted at the end of the prefilter module. Finally, the EM module takes as inputs the initial values for α, the compatibility matrix, and the target transcriptome index from the prefilter module. It estimates the transcript coverage abundances using an expectation-maximization (EM) algorithm. The EM algorithm converges when the total change in the relative transcript abundances (ρ) is less than a specified threshold, by default set to 0.05. The drop algorithm, described above and in Supplementary Figure S1, is implemented as a component of this module. It allows users to use the --drop flag to remove low compatibility relations between reads and transcripts immediately after the first E-step update. Read-to-transcript assignments (i.e., α estimates) and relative transcript abundances (i.e., ρ estimates) are outputted as TSV files at the end of the EM module. Users also have the option to further process the assignments and output hard 1-to-1 assignments between reads and transcripts for increased interpretability by specifying the --push flag.

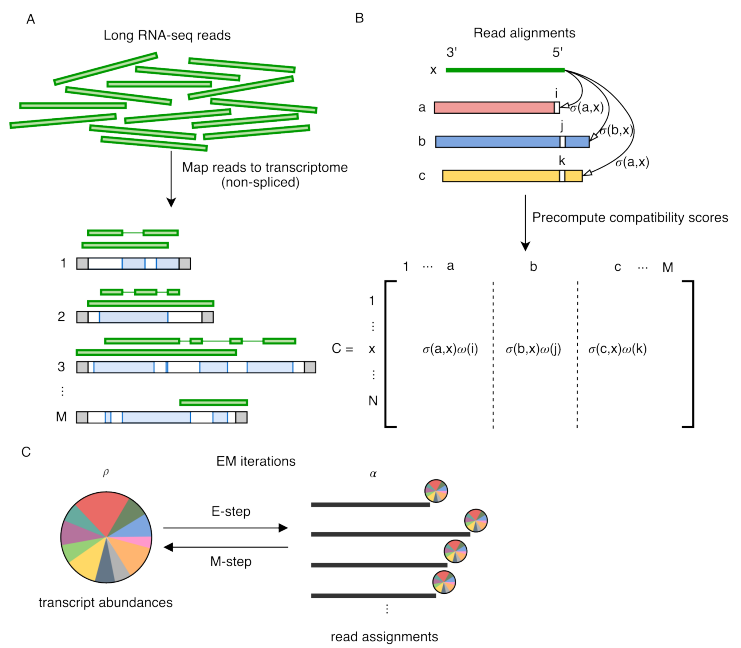

Figure 9. TranSigner's workflow consists of three modules: align, prefilter, and EM. A: In the align module, N reads are mapped to M transcript sequences; B: In the prefilter module, compatibility scores are precomputed, and some alignments are filtered out; C: In the EM module, read fractions are assigned to transcripts and transcript abundances are updated iteratively until convergence.

Simulated data. Three sets of Oxford Nanopore Technologies (ONT) dRNA reads and two sets of ONT cDNA reads were simulated using NanoSim (Gleeson et al., 2021). Expression levels were derived from protein-coding and long non-coding transcripts located on the main chromosomes (i.e., chromosomes 1 – 22, X, and Y) of the GRCh38 genome, extracted from the RefSeq annotation (release 110). We supplied the NA12878-dRNA

and -cDNA reads from Workman et al. to NanoSim's read characterization module to first construct two separate read profiles, one for generating dRNA and the other for generating cDNA reads. We then estimated the transcript abundances of the dRNA and cDNA samples by aligning each sample to the GRCh38 genome using minimap2 and providing the alignment results to salmon (Patro et al., 2017) in its alignment-based mode. These estimated transcript abundances were then used as input for the NanoSim simulation module. For each dRNA read set, we generated 14 million ONT dRNA reads, and 25 million for each cDNA read set.

Sequin data. We used an ONT direct-RNA dataset, which was released as part of the Singapore Nanopore Expression Project (SG-Nex) (Chen et al., 2021). This dataset was sequenced from the K-562 lymphoblast cell line and includes synthetic sequencing spike-in RNAs, also known as sequin RNAs. The identities and concentrations of the sequins were obtained from the SG-Nex AWS repository.

Read assignments evaluation. For simulated and sequin data, we can readily define the following values based on the known origin transcript of each read:

- True positive (TP): a read is correctly assigned to its true origin.
- False positive (FP): a read is incorrectly assigned to a transcript that is not its true origin.
- False negative (FN): a read is not assigned to its true origin.

If a read is assigned to multiple transcripts without specifying the fraction allocated to each transcript, then the read is evenly distributed among those transcripts, with these fractions contributing to TP and FP values as appropriate. If the exact fraction of a read assigned to a transcript is provided, those fractions are used instead.

For each sample, the recall value of a method for the read-to-transcript assignment is calculated as the number of TPs divided by the total number of reads sequenced from that sample. The precision value is computed as the number of TPs divided by the sum of TPs and FPs. F1 score is defined as 2 * precision * recall / (precision + recall).

Transcript abundance estimates evaluation. By default, TranSigner outputs read counts and relative transcript abundances as transcript abundance estimates. The read count of a transcript t (denoted as rc_t) is the sum of all positive read fractions assigned to transcript t, while the relative transcript abundance of t (denoted as ρ_t) is equal to rc_t normalized by the sum of all transcript read counts, ensuring that $\sum_{t \in T} \rho_t = 1$. Note that in a long-read RNA-seq experiment, each read counts as a transcript, making the sum of the read counts equivalent to the total number of transcripts identified from the long-read data.

TranSigner's read count estimates can be converted to counts per million (CPM) estimates by calculating $\mathrm{CPM}_t = \mathrm{rc}_t / l * 10^6$ where t is a transcript and l is the total number of reads (aligned and unaligned). TranSigner also outputs read-to-transcript assignments where each read is assigned to one or more transcripts. More precisely, TranSigner outputs a list of transcripts to which a read r is assigned along with the fraction of r is assigned to each transcript in that list, or the α estimates. These assignments can be used to compute coverage estimates for transcripts as $\lambda_t = \sum_{r \in R_t} \alpha_{rt} * (s_{rt} - e_{rt})$ where α_{rt} is the fraction of r assigned to transcript t, s_{rt} and e_{rt} are the alignment start and end positions of r in t, and R_t is the set of reads whose fractions were assigned to t. The alignment start and end positions are retrieved from the minimap2 alignment results saved during TranSigner's align module. If there is more than one alignment between a read r and the transcript t, then the alignment with the highest ms score is selected.

We performed both linear and nonlinear correlation analyses to evaluate the correlation between estimated and ground truth values, each assessing different qualities of the read assignment and quantification methods. While nonlinear correlation analysis, utilizing log-transformed read counts and Spearman's

correlation coefficient (SCC), evaluates monotonic trends in the data, linear correlation analysis, utilizing Pearson's correlation coefficient (PCC), assesses a tool's accuracy in assigning all reads to transcripts, valuing each read equally regardless of its source. It's worth noting that log transformation is typically applied to reduce variance in gene expression values. However, log transformation may compress differences in data points with large magnitudes, potentially diminishing the impact of errors in assigning reads to high abundance transcripts.

Evaluation of tools capable of transcriptome assembly. We assessed the quality of assemblies generated by StringTie, Bambu, and FLAIR using the intron chain-level sensitivity and precision values computed by GffCompare (Pertea & Pertea, 2020). We initially wanted to include ESPRESSO in this comparison, but we were unable to run it as it took more than 24h to process a single sample containing ~14 million reads.

We benchmarked each tool using random samples of the RefSeq annotation to observe how well the completeness of the guides impacts the accuracy of the assembled transcriptome. More precisely, we ran StringTie, Bambu, and FLAIR using 60 different guides on each of the three sets ONT dRNA simulated reads. StringTie and Bambu were provided with the same minimap2 alignment results produced using the recommended option for processing ONT direct RNA-seq reads (-x splice -uf -k14); FLAIR had its own align module. Unlike StringTie and FLAIR that output an annotation containing only the identified expressed transcripts, Bambu's output also contains all unexpressed transcripts in the guide annotation. Therefore, for our evaluations, we removed any transcript that was assigned a zero read count from Bambu's output.

The guides were sampled from the RefSeq annotation to contain 20 different percentages between 5 to 100% of the origin transcripts with increments of 5%. For each percentage, we independently sampled the annotation three times. The average assembly qualities across all percent guide annotations for all tools are shown in Figure 3. All precision, sensitivity, and F1 values reported below represent average values obtained across all read datasets using all guide samples for a given percentage of the origin transcripts. Bambu's precision and sensitivity values dropped drastically as the guide annotations became less complete, while StringTie's accuracy was the most resilient (see Supplementary Table S2).

When all origin transcripts were included in the guide annotation, Bambu achieved a sensitivity of 85.4% with a precision of 64.8%, while StringTie exhibited a lower sensitivity of 60.3% and a precision of 78.3%. However, with 50% of the origin transcripts included in the guide annotation, Bambu's sensitivity dropped to 60.1% with a precision of 48.6%, whereas StringTie showed a relatively smaller drop in assembly qualities and maintained a sensitivity of 54.1% with a precision of 72.7%. This was also reflected in an F1 score difference of 8.5 in favor of StringTie over Bambu. The gap in F1 scores between StringTie and Bambu widens as the guide annotations become increasingly incomplete. Both FLAIR and StringTie demonstrated relatively high precision values; however, StringTie slightly outperformed FLAIR. Bambu achieved better sensitivity than all other tools when more than 25% of the origin transcripts were present, but at 25% or less, StringTie surpassed Bambu. Conversely, when all origin transcripts were removed from the guide annotation (i.e., 0% guide annotations), StringTie outperformed Bambu and FLAIR by far in all three metrics with a relative increase of 80.9%, 20.7%, and 69.0% in sensitivity, precision, and F1 score compared to the next best performer. (see Supplementary Figure S3).

Data access

The *Arabidopsis thaliana* and Mouse data are available through ENA (PRJEB32782, PRJEB27590). ENA sample accession IDs for each pair of short- and long-read data sets are made available in Supplementary Table S9. The SG-NEx samples containing spike-in RNAs are available on GitHub, ENA, and AWS open data. The evaluation metric values used to generate figures in the main text are contained

in Supplementary Tables S2 ~ 8. The transcript abundances used to generate simulated ONT direct RNA-seq reads as well as the ground truth annotation, coverage, and read counts for each read set are made available in Zenodo at https://zenodo.org/doi/10.5281/zenodo.10966422. TranSigner is implemented in Python and is publicly available under the GPLv3 license at https://github.com/haydenji0731/TranSigner and is also archived in Zenodo at https://zenodo.org/doi/10.5281/zenodo.10967323.

Competing interest statement

The authors have declared no competing interest.

Acknowledgements

We would like to thank Beril Erdogdu for engaging in discussions on long-read RNA-seq models and Ales Varabyou for giving invaluable insightful into experimental setups. This work was supported in part by the US National Institutes of Health under grants R01-HG006677, and R01-MH123567. The funders had no role in the study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

Amaral, P., Carbonell-Sala, S., De La Vega, F. M., Faial, T., Frankish, A., Gingeras, T., Guigo, R., Harrow, J. L., Hatzigeorgiou, A. G., Johnson, R., Murphy, T. D., Pertea, M., Pruitt, K. D., Pujar, S., Takahashi, H., Ulitsky, I., Varabyou, A., Wells, C. A., Yandell, M., . . . Salzberg, S. L. (2023). The status of the human gene catalogue. *Nature*, *622*(7981), 41-47. https://doi.org/10.1038/s41586-023-06490-x

Amarasinghe, S. L., Su, S., Dong, X., Zappia, L., Ritchie, M. E., & Gouil, Q. (2020). Opportunities and challenges in long-read sequencing data analysis. *Genome Biol*, *21*(1), 30. https://doi.org/10.1186/s13059-020-1935-5

Benjamini, Y., & Speed, T. P. (2012). Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic acids research*, *40*(10), e72-e72.

Chen, Y., Davidson, N. M., Wan, Y. K., Patel, H., Yao, F., Low, H. M., Hendra, C., Watten, L., Sim, A., Sawyer, C., Iakovleva, V., Lee, P. L., Xin, L., Ng, H. E. V., Loo, J. M., Ong, X., Ng, H. Q. A., Wang, J., Koh, W. Q. C., . . . consortium, S.-N. (2021). A systematic benchmark of Nanopore long read RNA sequencing for transcript level analysis in human cell lines. *bioRxiv*, 2021.2004.2021.440736. https://doi.org/10.1101/2021.04.21.440736

Chen, Y., Sim, A., Wan, Y. K., Yeo, K., Lee, J. J. X., Ling, M. H., Love, M. I., & Göke, J. (2023). Context-aware transcript quantification from long-read RNA-seq data with Bambu. *Nature Methods*, *20*(8), 1187-1195. https://doi.org/10.1038/s41592-023-01908-w

Frankish, A., Diekhans, M., Ferreira, A. M., Johnson, R., Jungreis, I., Loveland, J., Mudge, J. M., Sisu, C., Wright, J., Armstrong, J., Barnes, I., Berry, A., Bignell, A., Carbonell Sala, S., Chrast, J., Cunningham, F., Di Domenico, T., Donaldson, S., Fiddes, I. T., . . . Flicek, P. (2019). GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res*, *47*(D1), D766-d773. https://doi.org/10.1093/nar/gky955

Gao, Y., Wang, F., Wang, R., Kutschera, E., Xu, Y., Xie, S., Wang, Y., Kadash-Edmondson, K. E., Lin, L., & Xing, Y. (2023). ESPRESSO: robust discovery and quantification of transcript isoforms from error-prone long-read RNA-seq data. *Science Advances*, *9*(3), eabq5072.

Gleeson, J., Leger, A., Prawer, Y. D. J., Lane, T. A., Harrison, P. J., Haerty, W., & Clark, M. B. (2021). Accurate expression quantification from nanopore direct RNA sequencing with NanoCount. *Nucleic acids research*, *50*(4), e19-e19. https://doi.org/10.1093/nar/gkab1129

Grünberger, F., Ferreira-Cerca, S., & Grohmann, D. (2022). Nanopore sequencing of RNA and cDNA molecules in Escherichia coli. *Rna*, *28*(3), 400-417. https://doi.org/10.1261/rna.078937.121

Hansen, K. D., Brenner, S. E., & Dudoit, S. (2010). Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic acids research*, *38*(12), e131-e131.

Kovaka, S., Zimin, A. V., Pertea, G. M., Razaghi, R., Salzberg, S. L., & Pertea, M. (2019). Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *Genome biology*, *20*, 1-13.

Li, B., Ruotti, V., Stewart, R. M., Thomson, J. A., & Dewey, C. N. (2009). RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, *26*(4), 493-500. https://doi.org/10.1093/bioinformatics/btp692

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics*, *34*(18), 3094-3100. https://doi.org/10.1093/bioinformatics/bty191

Li, H. (2021). New strategies to improve minimap2 alignment accuracy. *Bioinformatics*, *37*(23), 4572-4574. https://doi.org/10.1093/bioinformatics/btab705

O'Leary, N. A., Wright, M. W., Brister, J. R., Ciufo, S., Haddad, D., McVeigh, R., Rajput, B., Robbertse, B., Smith-White, B., Ako-Adjei, D., Astashyn, A., Badretdin, A., Bao, Y., Blinkova, O., Brover, V., Chetvernin, V., Choi, J., Cox, E., Ermolaeva, O., . . . Pruitt, K. D. (2016). Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res*, *44*(D1), D733-745. https://doi.org/10.1093/nar/gkv1189

Pachter, L. (2011). Models for transcript quantification from RNA-Seq. *arXiv preprint arXiv:1104.3889*.

Pardo-Palacios, F. J., Wang, D., Reese, F., Diekhans, M., Carbonell-Sala, S., Williams, B., Loveland, J. E., María, M. D., Adams, M. S., Balderrama-Gutierrez, G., Behera, A. K., Gonzalez, J. M., Hunt, T., Lagarde, J., Liang, C. E., Li, H., Meade, M. J., Amador, D. A. M., Prjibelski, A. D., . . . Brooks, A. N. (2023). Systematic assessment of long-read RNA-seq methods for transcript identification and quantification. *bioRxiv*, 2023.2007.2025.550582. https://doi.org/10.1101/2023.07.25.550582

Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., & Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, *14*(4), 417-419. https://doi.org/10.1038/nmeth.4197

Pertea, G., & Pertea, M. (2020). GFF Utilities: GffRead and GffCompare [version 2; peer review: 3 approved]. *F1000Research*, *9*(304). https://doi.org/10.12688/f1000research.23297.2

Tang, A. D., Soulette, C. M., van Baren, M. J., Hart, K., Hrabeta-Robinson, E., Wu, C. J., & Brooks, A. N. (2020). Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns. *Nature communications*, *11*(1), 1438.

Varabyou, A., Sommer, M. J., Erdogdu, B., Shinder, I., Minkin, I., Chao, K.-H., Park, S., Heinz, J., Pockrandt, C., Shumate, A., Rincon, N., Puiu, D., Steinegger, M., Salzberg, S. L., & Pertea, M. (2023). CHESS 3: an improved, comprehensive catalog of human genes and transcripts based on large-scale expression data, phylogenetic analysis, and protein structure. *Genome biology*, *24*(1), 249. https://doi.org/10.1186/s13059-023-03088-4

Wongsurawat, T., Jenjaroenpun, P., Wanchai, V., & Nookaew, I. (2022). Native RNA or cDNA Sequencing for Transcriptomic Analysis: A Case Study on Saccharomyces cerevisiae [Original Research]. *Frontiers in Bioengineering and Biotechnology*, *10*. https://doi.org/10.3389/fbioe.2022.842299

Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L., & Yorke, J. A. (2013). The MaSuRCA genome assembler. *Bioinformatics*, *29*(21), 2669-2677. https://doi.org/10.1093/bioinformatics/btt476