nature methods

Article

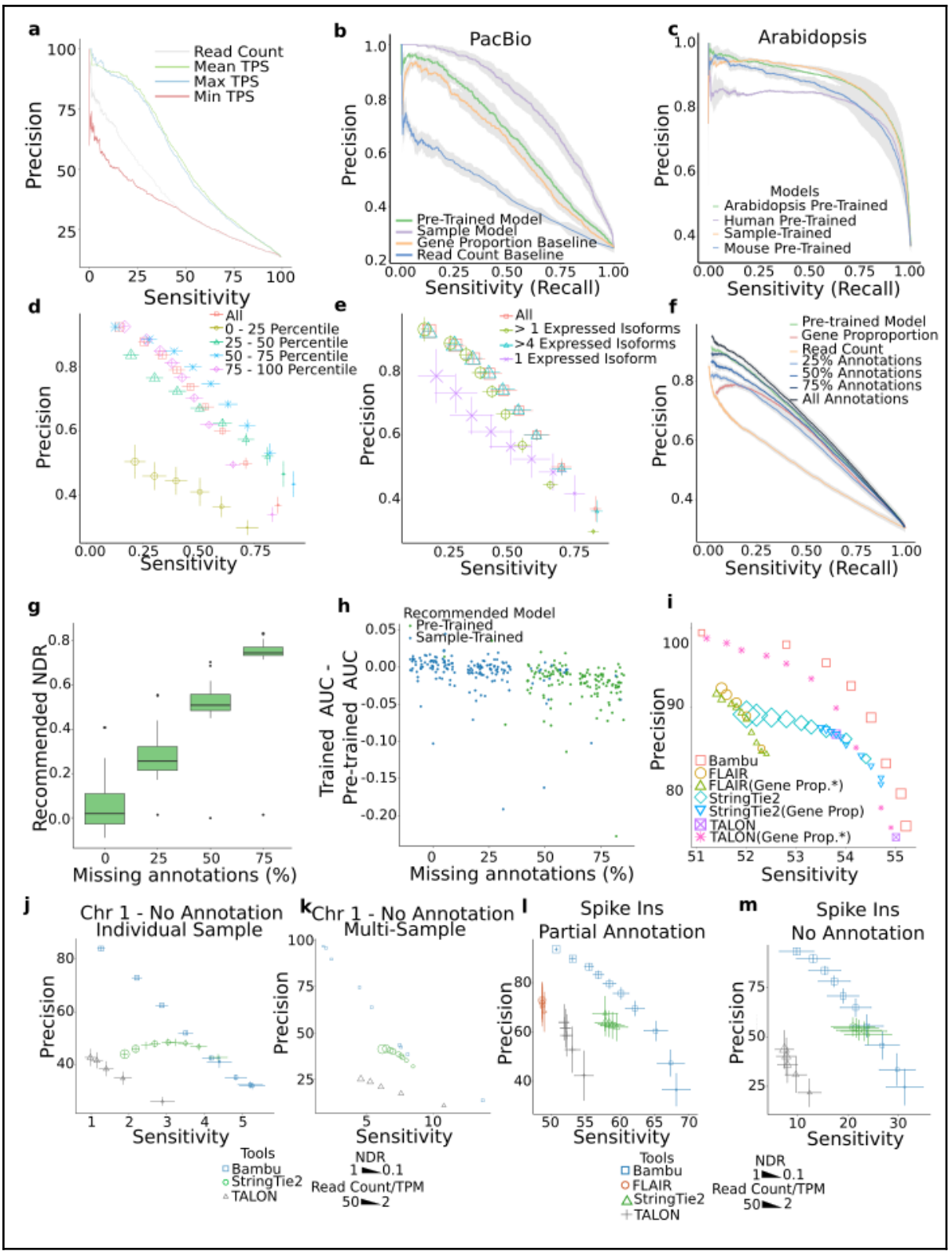
https://doi.org/10.1038/s41592-023-01908-w

Context-aware transcript quantification from long-read RNA-seq data with Bambu

In the format provided by the authors and unedited

Content

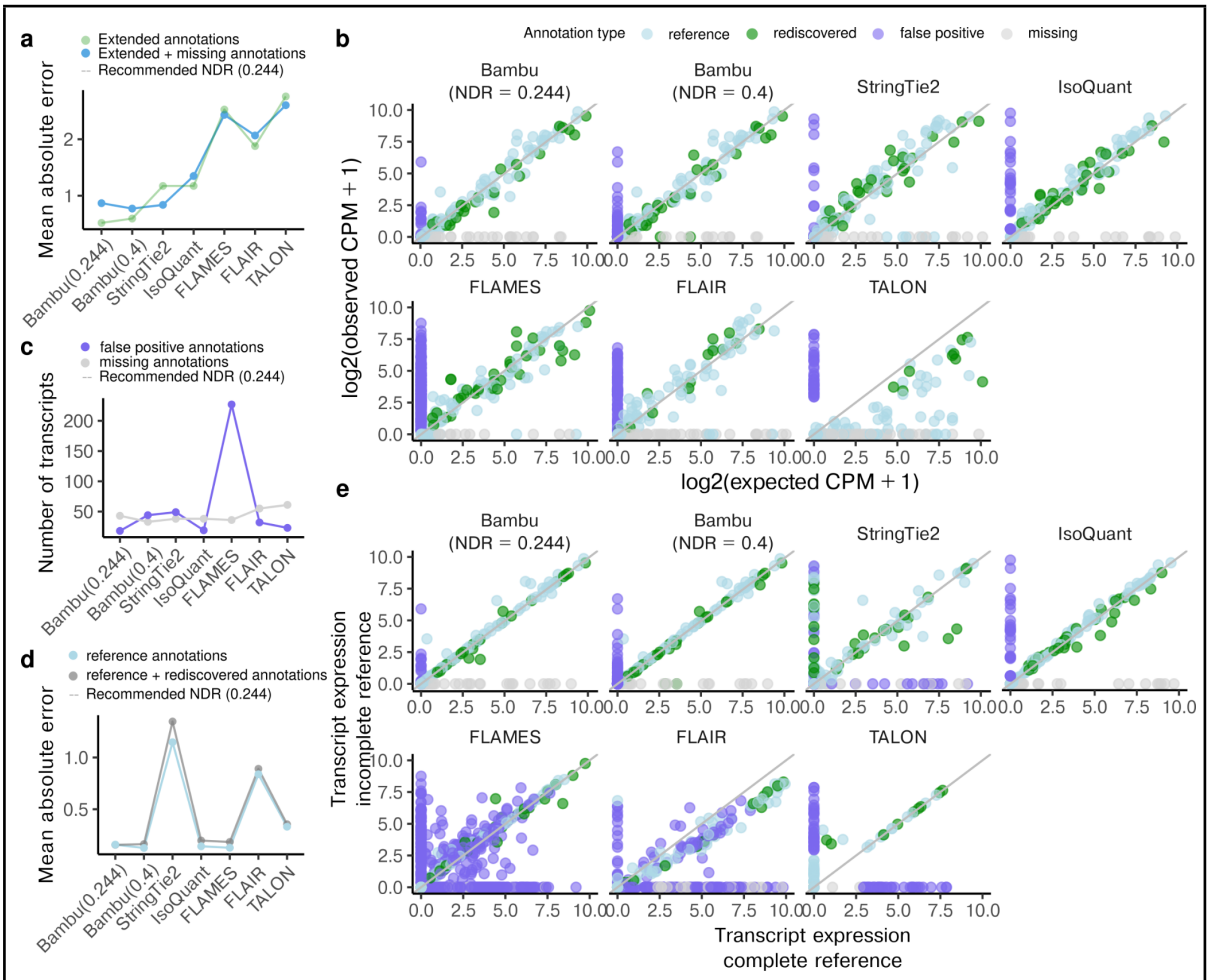
Supplementary Figures



Supplementary Figure 1. The transcript discovery model is robust and effective at classifying full-length read classes

(a) PR curves for the performance of transcript discovery when using minimum, mean, and maximum to combine TPS across samples for the same read class on all HepG2 samples together without annotations for human chromosome 1. The performance of using the sum of read counts as the classifier across the samples is used as a comparison **(b)** A precision recall curve showing the performance of Bambu on PacBio data using either the PacBio trained model (purple), the pretrained model (green), or ranking read classes by gene proportion (orange) or read count (blue). The grey shaded area represents the mean +/- SE of the precision for each line. **(c)** A precision recall curve showing the performance of models pre-trained on human (purple), mouse (blue) or arabidopsis (green) data applied to another arabidopsis tissue. Additionally the performance of the sample trained model is included (orange). The grey shaded area represents the mean +/- SE of the precision for each line. **(d)** The precision and sensitivity of varying Bambu thresholds when looking at a subset of read classes divided into expression quantiles. The same model is applied to all subsets. The full data is coloured in red, read classes that have expressions ranging from 0 to the lower quartile are shaded in yellow, those ranging between the lower quartile and the median in green, between the median and upper quartile in blue and the upper quartile to the max in red. Each subset should represent approximately 25% of the read classes. The lowest expressed quartile is larger than the others due to a greater than 25% of read classes sharing a read count of 2. **(e)** The precision and sensitivity of varying Bambu thresholds when looking at a subset of read classes divided by the number of expressed isoforms (> 2 read count) their gene contains. The same model is applied to all subsets. The full data is coloured in red, the subset of read classes that are the only expressed isoforms in their gene are coloured purple, those which have two or more isoforms are shaded in teal, and those with five or more isoforms are green. **(f)** Precision and Recall curves of the classification using Bambu models trained using reference annotations missing a random fraction of annotations used from the human reference annotations (excluding chromosome 1). The models trained using these annotations, are used to classify read classes from chromosome 1. The Pretrained Model represents the in-built model in Bambu which is used when the annotations do not support training and Read Count classifies the read classes solely using read count alone. These were applied to all SG-NEx datasets. **(g)** A box plot showing the distribution of NDR recommendations of SG-NEx samples (n = 76) when Bambu was run with different percentages of reference annotations. **(h)** We measured the difference in ROC AUC of trained and pretrained models in which the trained model was trained with missing reference annotations. Samples in which Bambu recommends using the pretrained model are coloured red (the recommended NDR was calculated as > 0.5), and samples where the trained model is used are coloured blue (**i**)The sensitivity and precision from transcript discovery on SGNex_HepG2_directRNA_replicate6_run1 with 50% of chr1 annotations randomly removed. Each tool is displayed at several different parameter thresholds. Bambu (red) was run using novel discovery thresholds between 1 and 0.1. StringTie2 (teal and blue) FLAIR (orange and green) and TALON (purple and pink) were run with read count/coverage thresholds between 2 and 10 and gene proportion thresholds between 0.01 and 07. FLAIR and TALON do not provide their own parameter for thresholding by gene proportion, so these thresholds were manually applied using the quantification results from the respective tool Error bars represent the standard error **(j)** The average sensitivity and precision of transcript discovery on core SG-NEx samples (n = 76) without annotations for human chromosome 1. Each tool is displayed at several different parameter thresholds: Bambu (blue) with NDR thresholds varying between 1 and 0.1, FLAIR (red), StringTie2 (green) and TALON (grey) with read count/coverage thresholds varying between 2 and 10, with 4 additional thresholds for StringTie2 at 15, 20, 30 and 50. Horizontal error bars represent the mean +/- SD of the sensitivity and vertical error bars represent the mean +/- SD of the precision **(k)** The measured sensitivity and precision of transcript discovery when combining HepG2 samples (n = 12), without annotations for human chromosome 1. Each tool is displayed at several different parameter thresholds: Bambu (blue) with NDR thresholds varied between 1 and 0.1, StringTie2 (green) and TALON (grey) with read count/coverage thresholds varying between 2 and 10, and 4 additional thresholds for StringTie2 at 15, 20, 30 and 50 **(l)** The average sensitivity and precision of transcript discovery on spike-in data (n = 8) with 50% of the spike-in annotations randomly removed. Each tool is displayed at several different parameter thresholds: Bambu (blue) with NDRthresholds varying between 1 and 0.1, StringTie2 (green), FLAIR (red) and TALON (grey) with read count/coverage thresholds varied between 2 and 10. Horizontal error bars represent the mean +/- SD of the sensitivity and vertical error bars represent the mean +/- SD of the precision **(m)** The average sensitivity and precision from transcript discovery outputs run on spike-in data (n = 8) without annotations for the spike-in chromosome. Each tool is displayed at several different parameter thresholds. Bambu (Blue) was run using

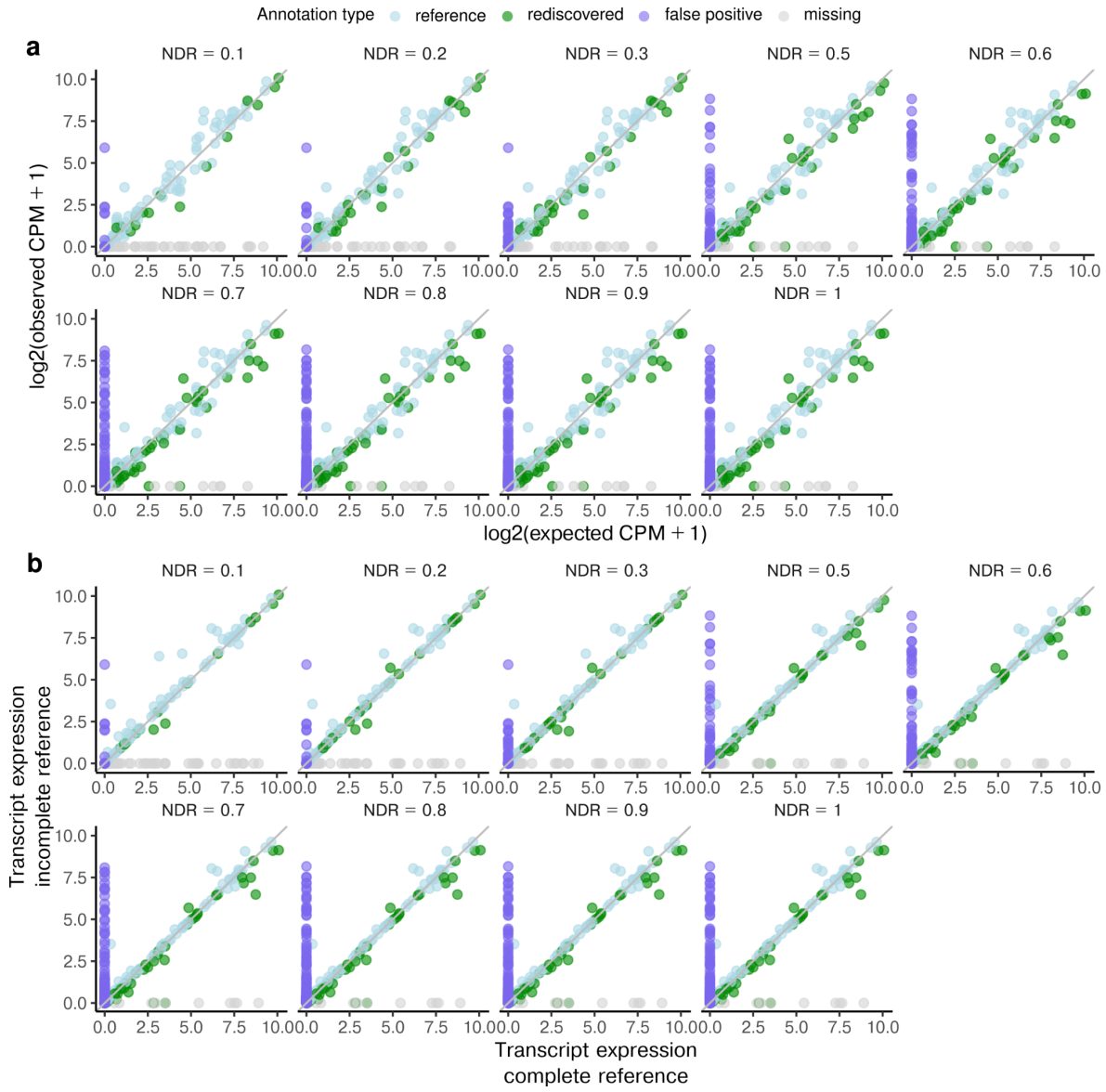
novel discovery thresholds between 1 and 0.1. StringTie2 (green), FLAIR (red) and TALON (grey) were run with read count/coverage thresholds between 2 and 10. Error bars represent the standard error



Supplementary Figure 2. Quantification accuracy and consistency by methods that does both transcript discovery and quantification

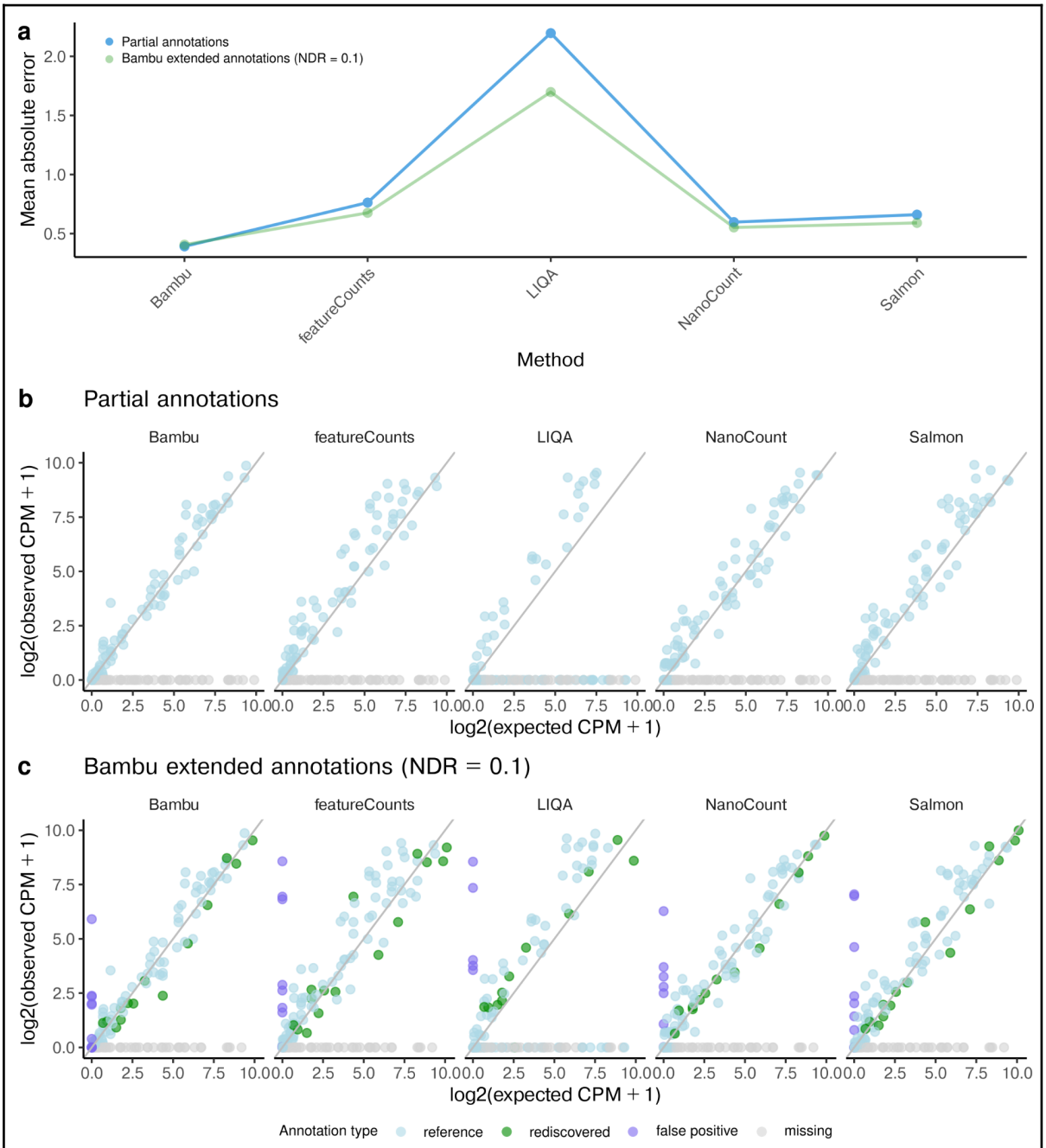
(a) The mean absolute error between the log2 normalised spike-in transcript abundance estimates and log2 normalised expected spike-in abundance when applying Bambu with recommended NDR (0.244), Bambu with NDR = 0.4, StringTie2, IsoQuant, FLAMES, FLAIR, and TALON for annotations for extended annotations by each of these methods, including annotations that are present in the reference (partial) sequin annotations, the annotations that have been artificially removed and rediscovered by each of the methods, and also the false positive annotations discovered by each method (green), plus annotations that have been artificially removed from the partial annotation and remained missing after transcript discovery, i.e., missing annotations (blue) **(b)** Scatterplots between log2 normalised transcript abundance estimates and log2 normalised expected spike-in abundance when applying Bambu with recommended NDR (0.244), Bambu with NDR = 0.4, StringTie2, IsoQuant, FLAMES, FLAIR, and TALON for spike-in transcripts annotated transcripts (light blue), transcripts artificially removed from the reference and rediscovered by Bambu (green), false positive transcripts (purple), transcripts artificially removed from the reference and remained missing after Bambu discovery (grey) **(c)**The number of missing (grey) and false positive (purple) transcripts using partial full sequin annotations when applying Bambu with default recommended NDR (0.244), Bambu with NDR = 0.4, StringTie2, IsoQuant, FLAMES, FLAIR, and TALON. **(d)** and full annotations when applying Bambu with recommended NDR (0.244), Bambu with NDR = 0.4, StringTie2, IsoQuant, FLAMES, FLAIR, and TALON. This is calculated separately for annotated transcripts (light blue), transcripts artificially removed from the reference (green), and false positive transcripts (purple) **(e)** Scatterplots between log2 normalised transcript abundance estimates with complete annotations and log2 normalised transcript abundance estimates with partial annotations when applying Bambu with recommended NDR (0.244), Bambu with NDR = 0.4, StringTie2, IsoQuant, FLAMES,

FLAIR, and TALON for spike-in transcripts annotated transcripts (light blue), transcripts artificially removed from the reference and rediscovered by Bambu (green), false positive transcripts (purple), transcripts artificially removed from the reference and remained missing after Bambu discovery (grey)



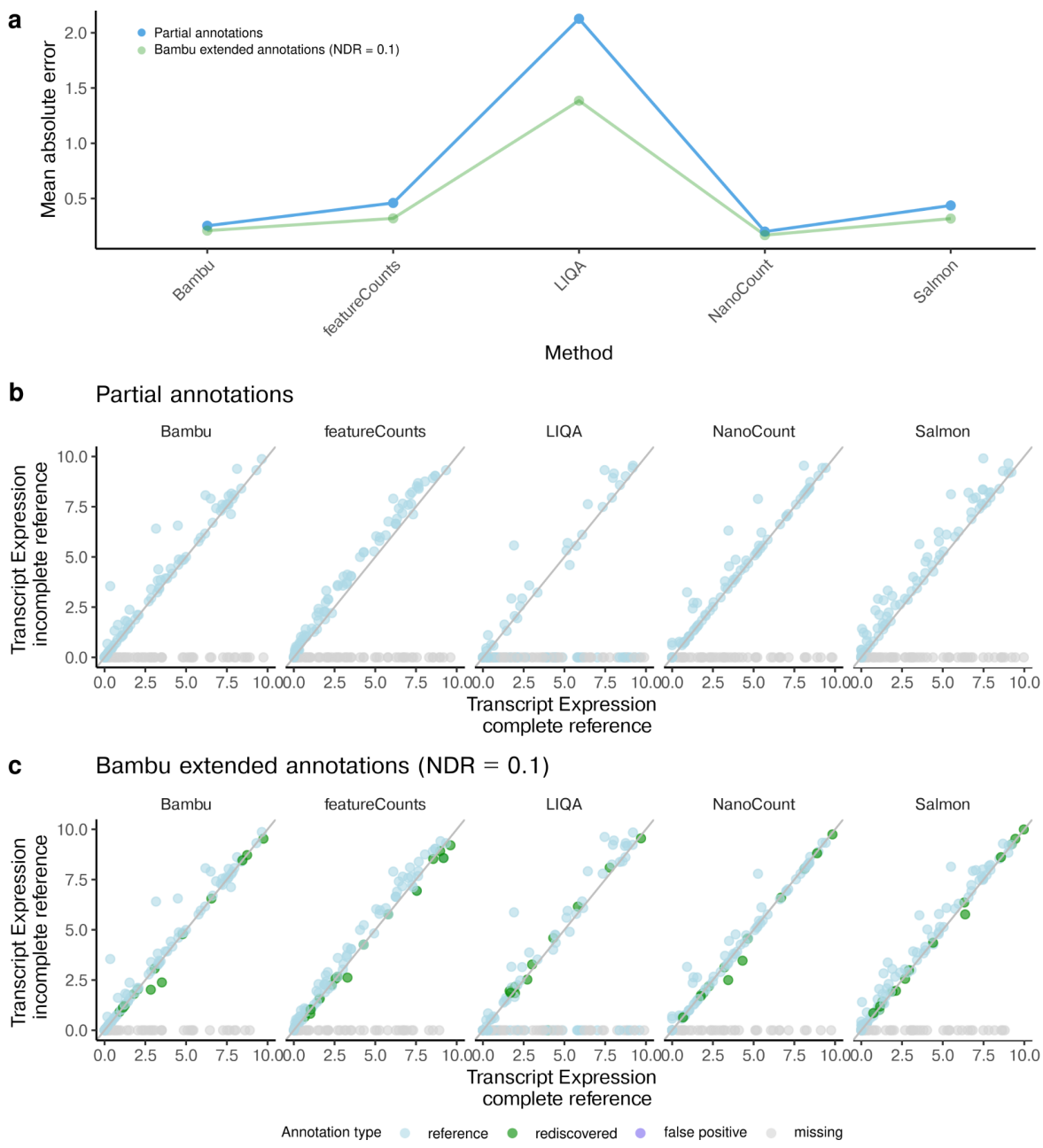
Supplementary Figure 3. Comparison of quantification for Bambu with other NDRs

(a) Scatterplots between log2 normalised transcript abundance estimates and expected spike-in abundance using partial annotations, and (b) Scatterplot between log2 normalised transcript abundance estimates using full and partial annotations: for spike-in annotated transcripts (light blue), transcripts artificially removed from the reference and rediscovered by Bambu (green), false positive transcripts (purple), transcripts artificially removed from the reference and remained missing after Bambu discovery (grey) when applying Bambu with varying NDRs from 0.1 to 0.3, and 0.5 to 1



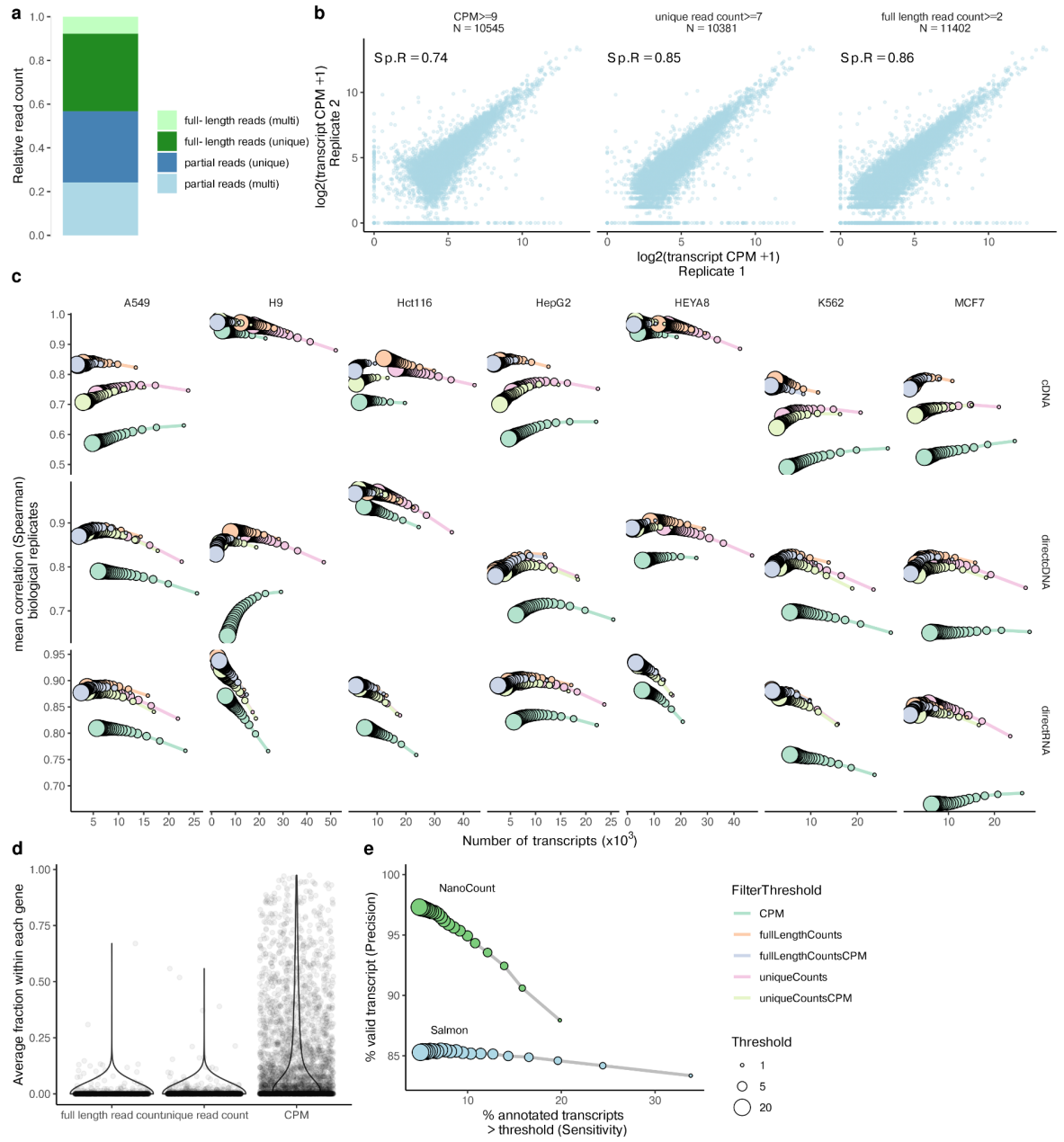
Supplementary Figure 4. Quantification performance when partial/extended annotations are provided

(a) The mean absolute error between the log2 normalised transcript abundance estimates and log2 normalised expected spike-in abundance when applying Bambu, featureCounts, LIQA, NanoCount, and Salmon with partial (blue) or Bambu extended (NDR = 0.1, green) annotations for transcripts that are present in the reference (partial sequin annotations). **(b-c)** Scatterplots between transcript abundance estimates and expected spike-in abundance when applying Bambu, featureCounts, LIQA, NanoCount, and Salmon for transcripts that are present in the reference (partial sequin annotations) (light blue), transcripts that have been artificially removed from the reference and rediscovered by Bambu (green), false positive transcripts (purple), transcripts that have been artificially removed from the reference and remained missing after Bambu discovery (grey): (b) with partial annotations; (c) with extended annotations by Bambu (NDR = 0.1)



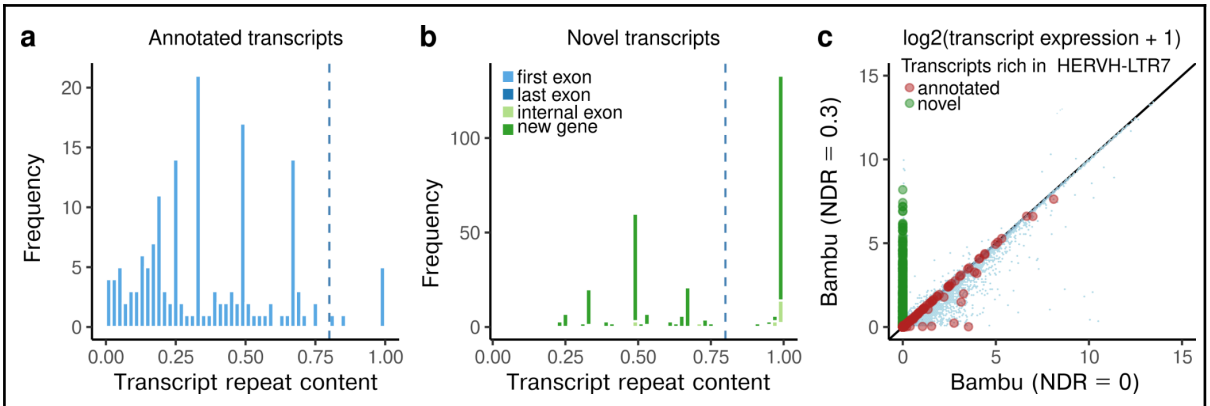
Supplementary Figure 5. Quantification when partial/extended annotations are provided compared against quantification when complete annotations are provided

(a) The mean absolute error between the log2 normalised spike-in transcript abundance estimates with partial (blue) or Bambu extended (NDR = 0.1, green) annotations against with full annotations when applying Bambu, featureCounts, LIQA, NanoCount, and Salmon for transcripts that are present in the reference (partial sequin annotations). **(b-c)** Scatterplots between log2 normalised transcript abundance estimates with complete annotations and log2 normalised transcript abundance estimates when applying Bambu, featureCounts, LIQA, NanoCount, and Salmon for transcripts that are present in the reference (partial sequin annotations) (light blue), transcripts that have been artificially removed from the reference and rediscovered by Bambu (green), false positive transcripts (purple), transcripts that have been artificially removed from the reference and remained missing after Bambu discovery (grey): (b) with partial annotations; (c) with extended annotations by Bambu (NDR = 0.1)



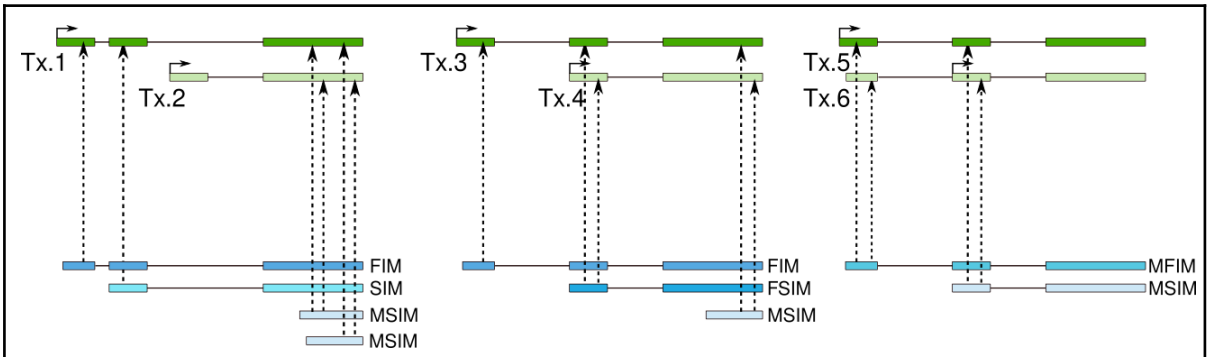
Supplementary Figure 6. Full-length and unique read support show evidence additional to transcript abundance estimates

(a) The distribution of average reads categorised as full-length, partial length, and unique **(b)** The comparison of transcript abundance estimates between two replicates of MCF7 generated using the direct cDNA protocol, with filtered transcript number being approximately 30 thousand, when filtered using CPM, unique read count, and full-length read count. The spearman correlation is shown in the top right for each filter **(c)** The mean spearman correlation of transcript abundance estimates between replicates for each cell line generated using cDNA, direct cDNA, and direct RNA protocols against number of transcripts that pass the filter, using CPM, full-length read count, full-length CPM, unique read count and unique CPM filter thresholds, with thresholds ranging from 1 to 20 **(d)** Violin plot showing the median, upper and lower quartiles, and 1.5 x interquartile range of the average fraction of full-length, unique read counts and CPM for each artificial transcript across Hct116 samples **(e)** The sensitivity and precision of NanoCount and Salmon abundance estimates in filtering transcripts overlapping with highly abundant isoforms that have no unique or full-length reads support at varying filtering thresholds from 1 to 20 on Hct116 samples. Filtering was based on the average values across replicates being not lower than the threshold



Supplementary Figure 7. Transcript discovery identifies novel transcripts overlapping highly with repeats

(a-b) Histogram of overlapping percentage of repeats for (a) annotated transcripts and (b) novel transcripts **(c)** Scatterplot of transcript abundance estimates with discovery (NDR = 0.3) against that without discovery (NDR = 0), with red and green points showing annotated and novel transcripts with at least 80% overlapping with HERVH-LTR7, and light blue points showing other transcripts with less than 80% overlapping with HERVH-LTR7



Supplementary Figure 8. Equivalence Read Class (EquiRC) types

Illustration of the five equiRC types: FIM (Full Intron Match), equally aligning to unique transcript; SIM (Subset Intron Match), partially aligning to unique transcript; FSIM (Full and Subset Intron Match), equally aligning to subset transcripts, while partially aligning to longer transcripts; MFIM (Multiple Full Intron Match), equally aligning to multiple transcripts, usually very similar transcripts; MSIM (Multiple Subset Intron Match), partially aligning to multiple transcripts (these are the mostly fragmented reads)

Supplementary Text

1. Features used for transcript discovery

Bambu's transcript discovery model uses nine features during classification: number of reads, gene proportion, the standard deviation of the starts and ends of read classes, and the number of polyAs and polyTs found at the read class start and ends, which are detailed below:

Feature definition:

Each read class i is described by a vector $x_i \in R^9$:

1. 1 Number of reads (x_i^{RC})

The number of reads for each read class is defined as the number of aligned reads that share the exact exon-junctions (c_i), normalised by library size for sample j (l_j):

$$x_i^{RC} = \frac{c_i}{l_j}$$

With $l_j = \sum_{i=1}^{M} c_i$.

This feature is an intuitive measure for the validity of a read class and its abundance in the sample of interest, as the read count directly reflects the number of observations for this read class in the data set.

Limitations of read count as a standalone parameter

Systematic errors during library preparation, sequencing, or alignment can result in larger read counts for read classes that are not valid transcripts, such as degraded RNAs and non-full-length reads. Furthermore, read classes with low read count can still be valid transcripts, and highly expressed genes are more likely to lead to read classes with high read count that originate from degradation products or other possible artefacts. Therefore, higher read count does not guarantee a high probability that a read class is a valid transcript, and a low read count does not guarantee that a read class is not a valid transcript.

1.2 Gene proportion (x_i^{GP})

Gene proportion represents the proportion of reads that are assigned to one read class amongst all the reads assigned to the same gene:

$$x_i^{GP} = \frac{c_i}{\sum_{t \in I} c_t}$$

With I representing the set of all read classes that overlap the same gene as read class i

Gene proportion directly addresses some of the main limitations of read count (x^{RC}) as it reflects the number of reads assigned to each read class relative to the number of reads assigned to the gene.

Limitations of read count as a standalone parameter

There are 2 main limitations for gene proportion. Firstly, this feature is strongly influenced by the number of isoforms for each gene and therefore not comparable across different genes. In particular, gene proportion is always 1 for single transcript genes, and much smaller with larger numbers of expressed transcripts. Secondly, the estimation of gene proportion can be inaccurate with low read counts. With a gene read count of 10, any read class with 2 reads will have a gene proportion of 20%, which is already higher than the gene proportion observed for many valid, annotated transcripts from genes with many isoforms.

1.3 TSS/TES standard deviation ($x_i^{\sigma TSS}$ and $x_i^{\sigma TES}$)

This feature is calculated by measuring the standard deviation of the locations of the start and end coordinates for all the reads comprising a read class. Differences in the standard deviation may indicate specific properties of valid transcripts that distinguish them from degradation artefacts and other invalid transcript candidates.

1.4 Strand bias (x_i^{strand})

The strand bias is calculated as the proportion of reads mapping to the strand with higher read count.

In cDNA samples sequencing can start at both the 5’ and 3’ end of the transcript (whereas in RNA samples, sequencing always starts at the 3’ end). For cDNA samples, a deviation from the average strand bias could represent cases where sequencing did not process correctly in one direction resulting in a systematic early truncation.

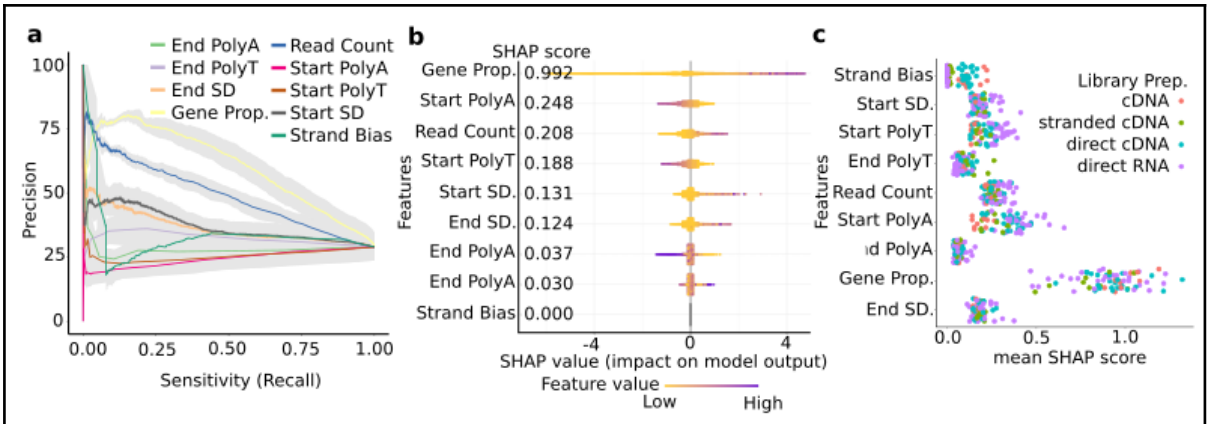
1.5 Start and End polyA/T frequency ($x_i^{start-A}, x_i^{start-T}, x_i^{end-A}, x_i^{end-T}$)

This feature counts the number of A’s (orT’s respectively) within the first (or last respectively) 10bp of the start and end of the read class. This feature is based on the genome sequence, not on the read or transcript sequence, and does not represent the presence of polyA tails. The presence of polyA/polyT sequences can result in early truncation of the read. One example is strand invasion that can occur during reverse transcription leading to truncated reads with an abundance of A’s at the 5’ end. This feature is designed to capture sequencing artefacts due to the presence of A or T rich sequences.

2. Contribution of transcript features to transcript discovery

Bambu trains a model that automatically optimises the contribution of all features for transcript discovery based on the sample and technology provided by the user. On their own, only read counts and gene proportion (and to a lesser amount the standard deviation of both ends) are useful as classifiers (Supplementary Text Figure 1a). However as Bambu can learn non-linear relationships, these features still contribute to the overall classification accuracy, with varying impact depending on the sample. In the sample used for the default pre-trained model, gene proportion is the most important feature, whilst the presence of polyA and polyTs at the start of the read class and the read count showed similar importance (Supplementary Text Figure 1b). As this sample is direct RNA and is therefore stranded, the strand bias feature has no relevance in the model, showing how training is able to adapt to the sample as needed. The relevance of the multi-feature approach is highlighted when looking at the change in contribution of the features across multiple samples and library preparations (Supplementary Text Figure 1c). For example the number of T's and A's at the start of a transcript have more impact on the direct RNA-seq data, reflecting specific characteristics of the direct RNA-Seq protocol.
The integration of features using a supervised machine learning model allows for more dynamic and context-specific transcript discovery. Besides being robust and accurate, this approach also reduces the complexity of threshold calibration from nine potentially relevant features to a single, interpretable probability score.

Bambu also allows users to specify read count and gene read proportion thresholds which can be used instead of the NDR, however, this is not recommended.



Supplementary Text Figure 1. Contribution of features to Bambu's transcript discovery model

(a) A precision-recall plot for each feature used in the transcript prediction model in bambu. Features are used to rank read classes/transcript candidates without any transformation. Each precision-recall curve is averaged across all SG-NEx data. The grey shaded area represents the standard error for each feature. **(b)** A SHAP plot showing the value of different features for a model trained on the HepG2 direct RNA replicate 5 run 1 sample. The y-axis is each feature used in the model with the number representing the feature SHAP score (the contribution of the feature) to the model. Each point represents one read class and the colour represents if the feature has a higher value (purple) relative to the feature. The x-axis is the importance of the features value to the model prediction score. **(c)** A jitter plot of the feature SHAP scores for all SG-NEx samples. Each feature from the model is on the y-axis and the mean SHAP score is shown on the x-axis. Each point is coloured based on the library preparation of the sample: PCR cDNA (red), PCR cDNA stranded

(green), direct cDNA (blue) and direct RNA (purple)

3. Combining Samples

Bambu trains a model on each sample individually and thereby assigns a different TPS to a read class that occurs across multiple samples. Bambu uses the maximum TPS to combine multiple samples (see Methods for details). As a comparison, we also measured the predictive power with a PR curve when either the maximum, minimum or mean is used to integrate the TPS across samples. Using the mean or max TPS resulted in the best and very similar performance with taking the minimum TPS performing worse than the read count baseline (Supplementary Figure 1a).

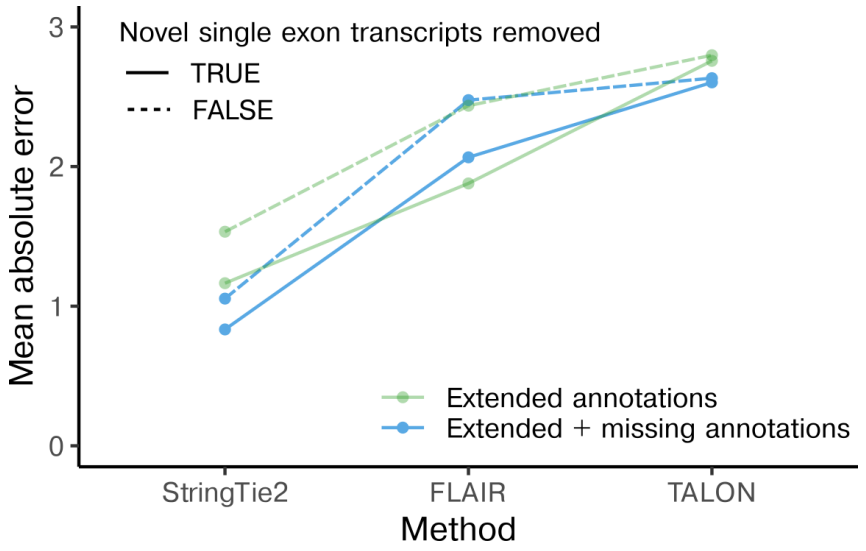
4. Single Exon Read Classes

Overlapping single exon reads are combined into single exon read classes. By default Bambu does not report single exon novel transcripts, however users can choose to include them if desired using advanced parameters (see online documentation).

During model training, bambu separately trains single-exon and multi-exon read classes and produces two distinct models. The read classes are not trained together as the features between the two read class types have differing behaviours which negatively affect model performance. Single exon read classes which are wholly contained within an annotated single exon annotation are considered as equal for training purposes (irrespective of how much the start and end sites differ). The TPS for multi-exon and single-exon read classes are predicted by their respectively trained models.

Single exon transcripts can be a source for false positives. While these transcripts can be included, we recommend that they are carefully inspected and interpreted. Single exon read classes are always used for quantification regardless of how they are handled during transcript discovery.

We have performed the quantification by other tools after removing the single exon transcripts. Compared to no filtering on novel single exon transcripts , the number of false novel transcripts is reduced for StringTie2, FLAIR, and TALON, reducing the mean absolute error (Supplementary Text Figure 2, other methods did not report single exon transcripts for the sequin genes).



Supplementary Text Figure 2. Impact of novel single exon transcripts on quantification accuracy

The mean absolute error between the log2 normalised spike-in transcripts estimates and log2 normalised true concentration levels when applying StringTie2, FLAIR, and TALON with (solid line) or without (dashed line) novel single exon transcripts removed for software annotations, i.e., partial annotations + positive annotations from each of the methods (green), or complete annotations, i.e., full reference annotations + positive annotations for each of the methods (blue).

5. Using a pre-trained model (in practice)

The pretrained model is shown to be able to effectively classify novel transcripts when tested on data from different technology (PacBio) and on different organisms (Arabidopsis) with only a small performance drop compared to a sample trained model (Supplementary Figure 1b). However, for users that do not have sufficient annotations to train their sample, and would like to increase the performance of transcript discovery, Bambu allows for training new models using related data for which comprehensive reference annotations are available. To evaluate this functionality, we trained Bambu on mouse, human and Arabidopsis data, and tested the pre-trained models on a different genotype of Arabidopsis. Here we observe that both the human and mouse pre-trained models can be used to rank novel transcripts in Arabidopsis (Supplementary Figure 1c). However, the model that was trained on *A. Thaliana* and applied to another genotype of *A. Thaliana* showed higher performance that was comparable with the sample/genotype specific model (Supplementary Figure 1c). These results suggest that the generic pre-trained model is robust and can be used effectively, while a pre-trained model that matches the species of interest is expected to lead to an improved performance.

Please refer to the online documentation (https://github.com/GoekeLab/bambu) for details on model training.

6. Bambu performance at different levels of annotation completeness

Reference annotations	Mean number of expressed annotated transcripts	Mean Model performance. (PR AUC)	Mean ROC AUC
25%	2345	0.597	0.770
50%	4688	0.639	0.785
75%	6999	0.663	0.792
100%	9260	0.677	0.798
Pretrained Model	NA	0.668	0.792
Read Count	NA	0.498	0.678

Supplementary Text Table 1. Performance of Transcript Discovery Model trained with missing reference annotations

Reference annotations represent the random fraction of annotations used from the human reference annotations (excluding chromosome 1). The models trained using these annotations, are used to classify read classes from chromosome 1. The Pretrained Model represents the in-built model in Bambu which is used when the annotations do not support training and included chromosome 1 during training. Read Count classifies the read classes solely using read count alone. These were applied to all SG-NEx datasets.

Missing Annotations	0%	25%	50%	75%	Number of Samples
PacBio Human	0.141 ±0.004	0.349 ±0.009	0.569 ±0.001	0.766 ±0.006	3
Mouse	0.128 ±0.010	0.336 ±0.010	0.557 ±0.015	0.774 ±0.004	4
Arabidopsis	0.103 ±0.007	0.309 ±0.009	0.527 ±0.002	0.755 ± 0.002	3

Supplementary Text Table 2. NDR recommendation on different datasets

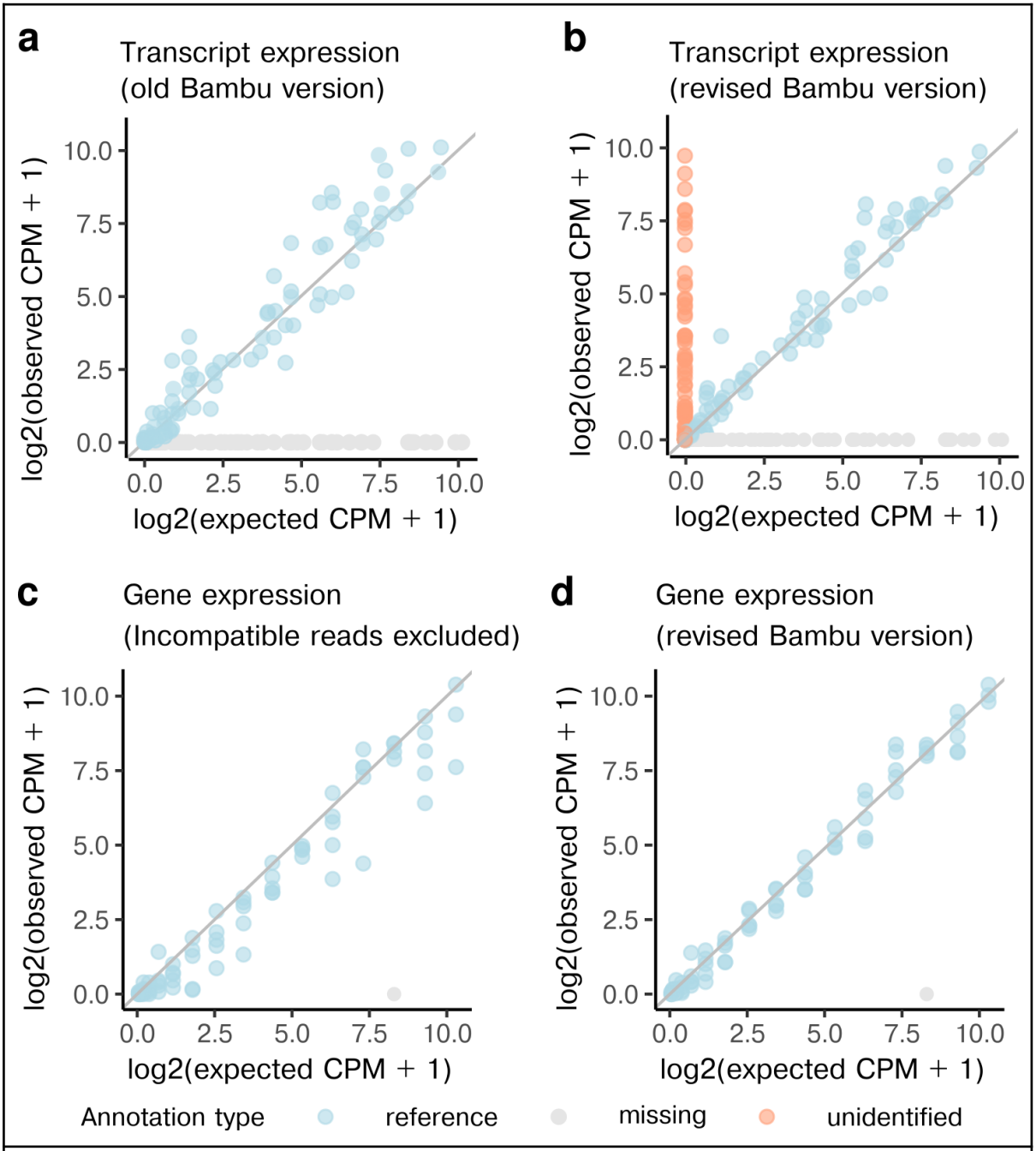
The table shows the mean recommended NDR across the samples tested. ± is the standard deviation of the mean

7. Filtering incompatible read classes for transcript quantification

To improve Bambu quantification accuracy, we only assign reads to transcripts if they are below a maximum alignment distance of 35bps (referred to as *compatible*), whereas any other read will be excluded from transcript quantification (*incompatible* reads). The filtering step provides more accurate transcript quantification (Supplementary Text Figure 3a-b).

Most reads which are incompatible with transcripts due to missing annotations can still be accurately assigned to genes. Usually, gene expression is estimated as the sum of transcript expression, therefore, when incompatible reads are removed, gene expression will be

underestimated. To prevent this, Bambu assigns all incompatible reads to an artificial "unidentified transcript" that is associated with each gene. Gene expression is then estimated using all reads that can be assigned to the transcripts of each gene, including reads that are incompatible with all existing annotations, leading to improved gene expression quantification (Supplementary Text Figure 3c-d).



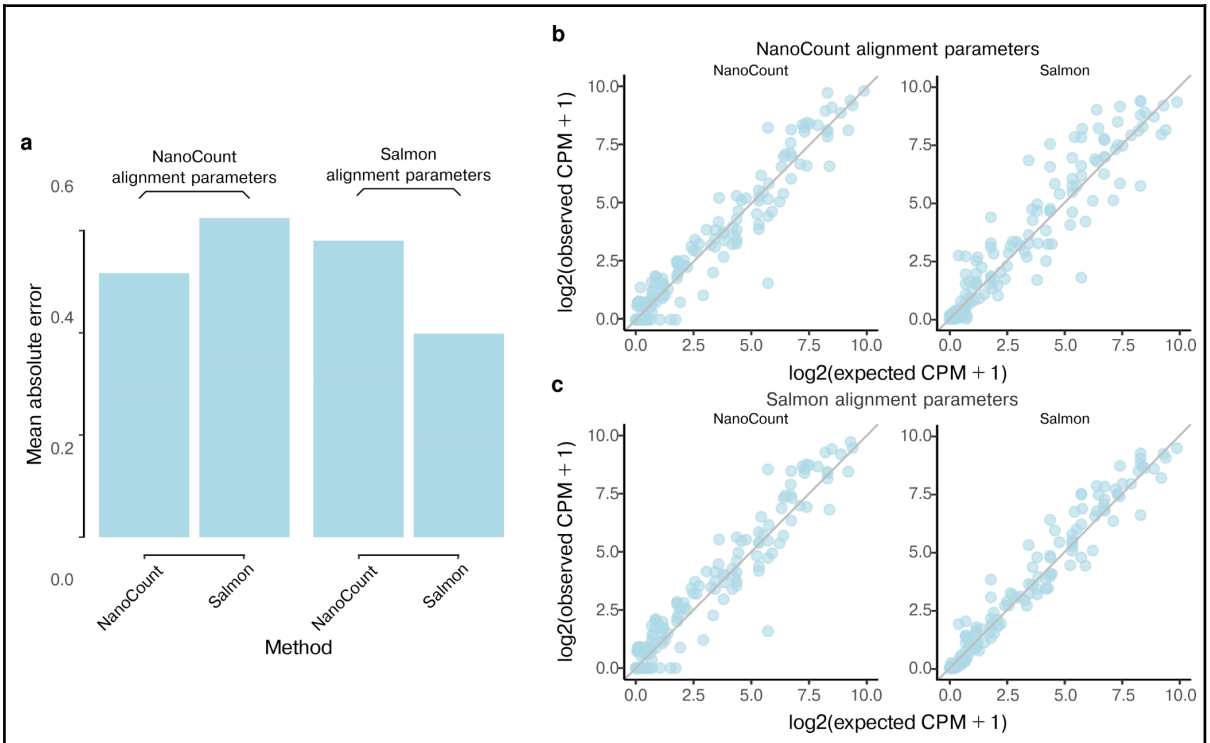
Supplementary Text Figure 3. Tracking of incompatible reads improves gene expression quantification

Shown is the scatter plot of observed CPM vs expected CPM for sequin transcripts and genes when Bambu is used with NDR = 0 and partial annotation is provided: **(a)** sequin transcript expression (old Bambu version) vs **(b)** sequin transcript expression (revised Bambu version); **(c)** sequin gene expression when incompatible reads are excluded vs **(d)** sequin gene expression when incompatible reads are included (revised version of Bambu). Blue dots represent transcripts

that are present in the partial annotation. Grey dots represent transcripts that are artificially removed from annotation, i.e., the missing transcripts in the partial annotation. Orange dots represent the unidentified transcript expression for each gene, with each dot representing one gene. Unidentified transcripts are only used for gene expression estimates, but not for transcript expression estimates, leading to improved quantification.

8. Impact of Minimap2 alignment parameters used on NanoCount and Salmon quantification results

For the transcriptome alignment-based methods NanoCount and Salmon, we aligned fastq files to the transcriptome with the recommended/default alignment steps that differ in the number of alignments reported for each read [1,2](see Supplementary Text Table 3 for the detailed parameter settings). To assess the impact of alignment parameters on quantification, we additionally applied NanoCount with alignments generated using Salmon recommended alignment parameters, and Salmon with alignments generated using NanoCount recommended parameters (Supplementary Text Figure 4). We find that both NanoCount and Salmon quantification results for the spike-in transcripts are similar when different alignment parameters are used (Supplementary Text Figure 4). For both methods, the recommended alignment parameters give better results, therefore we have kept the different alignment settings for the transcriptome-alignment based methods.



Supplementary Text Figure 4. The impact of transcriptome alignment parameters on quantification for NanoCount and Salmon

(a) Shown the barplot of mean absolute error between log2 normalised spike-in transcript abundance estimates and log2 normalised expected abundance when applying NanoCount and

Salmon with NanoCount recommended alignments parameters vs with Salmon recommended alignment parameters **(b)-(c)**Shown the scatterplots of log2 normalised spike-in transcript abundance estimates vs log2 normalised expected abundance when applying NanoCount and Salmon with (b) NanoCount recommended alignment parameters and (c) Salmon recommended alignment parameters are used.

Method	Minimap2 alignment parameters
NanoCount*	" -ax map-ont -p 0 -N 10"
Salmon*	"-ax map-ont -p 1.0 -N 100"
FLAMES**	"-ax splice -t 12 -k14" and "-ax map-ont -p 0.9 --end-bonus 10 -N 3"
All other methods	"-ax splice --junc-bed -k14", "-uf" for stranded samples

Supplementary Text Table 3. minimap2 alignment parameters

*NanoCount and Salmon aligned fastq files to transcriptome fasta file

**FLAMES did alignment twice, once to genome fasta file with "-ax splice -t 12 -k14" and once to transcriptome fasta file with "-ax map-ont -p 0.9 --end-bonus 10 -N 3"

9. Benchmark on running time and memory usage

Being able to analyse larger sample numbers with reasonable running times was a key consideration during the design and implementation of Bambu. To achieve this, we have vectorised most computation steps for efficient calculation in R, and we have implemented parallel processing and memory friendly file handling. We specifically evaluated the following scenarios for a systematic comparison the running time and memory of Bambu with existing methods for transcript discovery and quantification

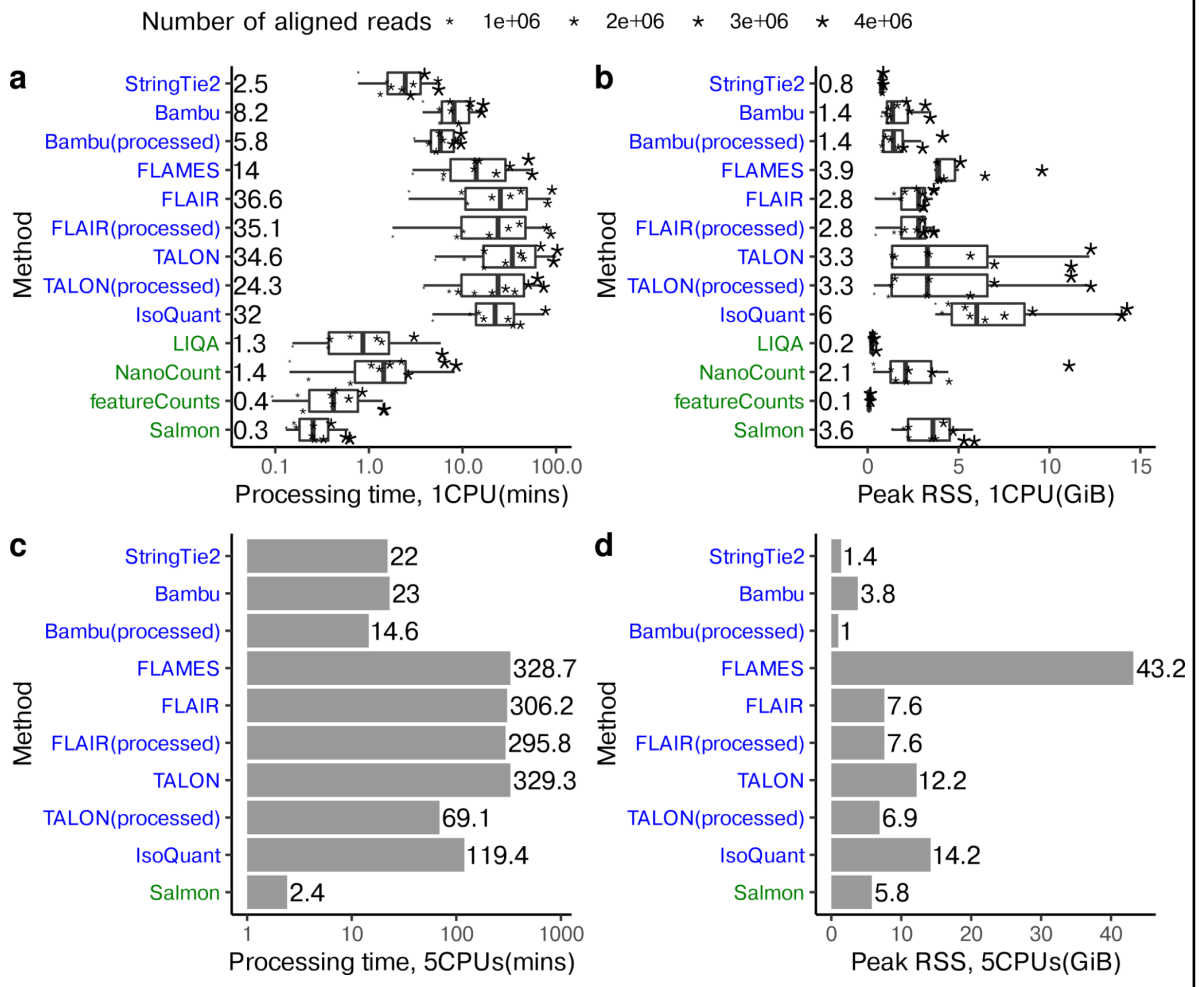
(1) Processing of individual samples

For this evaluation, we used 10 samples with varying sequencing depth (500K to 4.5 million reads) that were processed individually using a single CPU. Compared to other transcript discovery methods, Bambu is the second most efficient (average running time of 8.15 minutes and 1.4 GB RAM) after StringTie2 (2.45 minutes, 0.77GB RAM), with all other methods having running times between 14 and 32 minutes (Supplementary Text Figure 5).

For completeness we have also included methods that only perform quantification. As expected, these methods are generally faster as they do not attempt to identify novel transcripts (Supplementary Text Figure 5).

(2) Parallel processing of multiple samples

Most use cases of Bambu will include multiple samples that are jointly analysed. To compare the running time and memory usage of Bambu with other methods in this scenario, we used the same 10 samples and processed them as a single analysis using 5 CPUs. For this analysis, only methods that allow customised multi-threading were included. Here, Bambu has a running time of 23 minutes, which is comparable to StringTie2 (21 minutes), and significantly faster than any other transcript discovery method (295-330 minutes, Figure Supplementary Text Figure 5). Bambu supports the re-analysis with pre-processed files (for example when new samples are added to an existing analysis, or alternative thresholds are tested). Using this option further reduces the running time to 14.5 minutes.



Supplementary Text Figure 5. Comparison of processing time and peak RSS usage

Shown is the processing time for 10 samples when **(a)** processed individually with 1 CPU and **(c)** processed together with 5 CPUs; and the processing peak resident set size (RSS) usage for 10 samples when **(b)** processed individually with 1 CPU and **(d)** processed together with 5 CPUs. Methods that perform both transcript discovery and quantification have names colored in blue, while methods that only perform transcript quantification have names colored in green. Note that FLAMES can process multiple samples when fastq files are provided. IsoQuant is unable to process all samples together as not all samples include the spike-in chromosome. All other

methods were able to process the data. For IsoQuant, we processed the 10 samples individually with 5 CPUs as an approximation.

(3) Ability to process large data sets (121 million reads)

Due to improvements in the sequencing chemistry, an increasing number of reads is expected from long read RNA-Seq data. We therefore also evaluated the ability to perform transcript discovery and quantification on a sample with very high throughput (121 million reads in total, 94.6 million aligned reads). Bambu can successfully analyse such samples (Supplementary Text Table 4), whereas TALON, IsoQuant, NanoCount and LIQA either return errors or have running times of > 4 days.

	Method	Processing time (mins)	Peak RAM (GiB)	Number of cpus used*
Transcript discovery and quantification methods	Bambu	98.11	43.01	1
	StringTie2	87.80	1.84	12
	FLAIR	701.41	7.74	12
	FLAMES	1458.87	97.63	3
Transcript quantification methods	featureCounts	31	0.07	1
	Salmon	4.98	12.42	24

Supplementary Text Table 4. Processing time and peak memory usage for a very large sample (121 million reads).

Only methods that completed the analysis are shown.

*different cpus are used here to quickly process the sample. For featureCounts, no multi-thread is allowed. For Bambu multi-threading is most efficient when multiple samples are provided to facilitate memory-efficient compute, here only a single CPU is used. For flames three CPUs are used to prevent large memory usage

(4) Ease-of-use

Efficiency not only implies low running time and memory, but also ease of use. Bambu was designed to simplify the analysis of long read RNA-Seq data, minimising the number of commands and user-defined thresholds to obtain transcript annotation and quantification results. Among the transcript discovery methods, Bambu, FLAMES, and IsoQuant only require a single command to perform transcript discovery and quantification of 10 samples (Supplementary Text Table 5). In contrast, other methods analyse samples individually before combining them, requiring 10 to 12 commands to obtain the results from such a

multi-sample analysis (Supplementary Text Table 5). Compared to FLAMES and IsoQuant, Bambu is significantly faster (see above).

Method		**Number of commands for 10 samples (ease of use)**	
		discovery	**quantification**
Transcript discovery and quantification	*Bambu*	1 command	
	FLAIR	10 Correct + 1 Collapse (11 commands)	Quant 1 command
	TALON	1 Initialise DB + 10 talon_label_reads + 1 discovery (12 commands)	Quant 1 command
	StringTie2	10 Discovery + 1 merge (11 commands)	Need to re-run StringTie again with “-B -e” parameters (10 commands)
	FLAMES	1 command	
	IsoQuant	1 command	
Transcript quantification only	*LIQA*	Not applicable	10 commands
	NanoCount	Not applicable	10 commands
	featureCounts	Not applicable	10 commands
	Salmon	Not applicable	10 commands

Supplementary Text Table 5. Ease of use in transcript discovery and quantification when processing 10 samples

10. Feature comparison

In this section, we highlight the novel aspects for quantification of long read RNA-Seq in Bambu and present them in a table (Supplementary Text Table 6).

Method provide long read quantification	Able to quantify gene expression	Able to process multiple samples	No extra steps need to match annotations	Able to track full-length and unique read count estimates	Able to process very large sample (over 100 million reads, using 96 processors and 186.7GiB RAM)
Bambu	✔	✔	✔	✔	✔
NanoCount	✘	✘	✔	✘	✘
Salmon	✔*	✘	✔	✘	✔
featureCounts	✔*	✘	✔	✘	✔

StringTie2	✔*	❌	✔	❌	✔
FLAIR	✔*	✔	❌	❌	✔
TALON	✔	✔	✔	❌	❌
LIQA	❌	✔	✔	❌	✔**
FLAMES	✔*	✔	✔	❌	✔
IsoQuant	✔*	✔	✔	❌	❌

Supplementary Text Table 6. Feature comparison of methods that provide transcript quantification for long read data

✔: Possible
❌: Not possible
* not using incompatible reads;
**The whole process takes about 4 days.

11. Software versions

In this section, we listed out the versions of the softwares that were used in the benchmark analysis.

Software	Version used in transcript discovery benchmark	Version used in transcript quantification benchmark
Bambu	BambuManuscriptRevision branch	BambuManuscriptRevision branch
StringTie2	2.1.5	2.1.7
FLAIR	1.5.1	1.4 using docker, the docker version is not updated since 2019 Jun 26
TALON	5.0	5.0
IsoQuant	3.1.2	3.1.2
LIQA	NA	1.1.16
FLAMES	NA	0.1.0
NanoCount	NA	1.0.0.post3
Salmon	NA	1.9.0
featureCounts	NA	Rsubread_2.12.2

Supplementary Text Table 7. Versioning information for all methods benchmarked

Supplementary Notes

1. Transcript discovery evaluation

To evaluate the accuracy of the model in identifying valid novel transcript candidates we designated the read classes derived from spike-ins and chromosome 1 as the test set and the remaining read classes as the training set. The model was trained for each sample separately on all SG-NEx spike-in data (n = 8) and all core SG-NEx data (n = 76) (Supplementary Table 1). Sensitivity is measured as a percentage of the expressed annotated transcripts in the sample (read classes that are full-splice-matches to the reference annotation).

To measure the interpretability of the NDR threshold, we ran Bambu without the annotations for chromosome 1 on all core SG-NEx samples (n = 76) (Supplementary Table 1). We then measured the precision as the fraction of read classes from chromosome 1 that were below the varying NDR thresholds matching the splice junctions of reference annotations from chromosome 1. This was similarly performed for different read count thresholds. We also compared the precision of Bambu to both StringTie2 [3] and TALON [4] across all core SG-NEx samples (n = 76) (Supplementary Table 1), which are of varying sequencing depths. To match the default read count threshold used in Bambu, StringTie2 was run with the recommended “-L -G " parameters as well as “-c” at 2 which represents the minimum reads per bp coverage to consider for multi-exon transcript. Similarly, TALON was run withdefault parameters except for --minCount at 2 as part of talon_filter_transcripts which represents the number of minimum occurrences (reads) required for a novel transcript.

To benchmark the performance of Bambu in transcript discovery, we generated a partial annotation where we randomly removed 50% of the annotations on chromosome 1. We then

ran transcript discovery with Bambu, FLAIR [5], StringTie2, IsoQuant [6] and TALON with default parameters where applicable using all core SG-NEx samples (n = 76) (Supplementary Table 1). For Bambu, we varied the NDR threshold from 0.1 to 1 (0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1), For StringTie2 we varied the "-c" parameter choosing values between 2 and 50 (2, 4, 6, 8, 10, 15, 20, 30, 50). IsoQuant (green) was run using the "--model_construction_strategy" parameter of reliable default_ont and sensitive_ont. Due to the higher running time for TALON and FLAIR, we only varied "--minCount" and "-s" in as part of "flair collapse" which represents the minimum number of supporting reads for an isoform from 2 to 10 with a step of 2 respectively. All the novel isoforms were combined with the partial annotations to provide the final output annotation. For each tool, we then evaluated the sensitivity and precision using gffCompare [7] with default parameters by comparing the final output annotation with the complete annotations of chromosome 1. gffCompare measured sensitivity as the proportion of transcripts in the reference annotations that were detected. As 50% of the annotations of chromosome 1 are provided, a minimum of 50% is expected. As starts and ends are usually very challenging to determine, we focused on the intron chain level performance which also ignores the presence of single exon transcripts in all tools. To test the performance of these tools across multiple samples, we repeated the above analysis but with all SG-NEx HepG2 samples together (n=12) (Supplementary Table 1). TALON was excluded from this analysis as we could not successfully run the tool using these samples. For StringTie2, we used "--merge" to combine the output annotations from all samples at the different "-c" thresholds described above. StringTie2 was additionally run using "-T" which represents the minimum input transcript-per-million to include in the merge and was performed on the isoforms discovered at "-c 2".

2. Transcript quantification with context-specific annotations

To assess the impact of context-specific annotations on quantification, we generated a partial annotation for the Sequin chromosome [8,9] by removing 1 transcript at random for each multiple-isoform gene, and removing single-isoform genes at a random 50% probability. By doing so, we expected that the partial annotation would be missing 40-50% of its transcripts. The partial sequin annotations were then added to the Grch38 Ensembl annotations release version 91[10] . We varied the novel discovery rates (NDRs) from 0 to 1 with a 10% increasing gap and applied it on the samples with Sequins (MixA V2) in the SG-NEx core cell line samples (HEYA8 samples, n = 10). The estimated transcript expression levels are then compared against the expected transcript expression levels per million, calculated by the relative concentration levels times the expected number of Sequin reads per million (1% x 1 million, where 1% is the spike-in percentage), and the estimates when the Sequin chromosome is provided at full. For the full annotation analysis, we applied Bambu without discovery, i.e., NDR = 0.

We benchmarked Bambu against transcript-discovery assisted quantification approaches, StringTie2, FLAIR, TALON, and quantification-only approaches, LIQA [11], NanoCount [12], Salmon [13] and featureCounts. Genome bam files and unmapped fastq files are used as input per condition as described below. For StringTie2, we first performed StringTie2 with the recommended "-L -G" parameters on bam files to discover novel transcripts in each sample. We then performed " stringtie --merge " to combine novel transcripts across samples. Lastly, we repeated the first step with "-B -e" parameters added and the merged gtf following " -G " parameter to quantify the transcripts across samples. For FLAIR, we generated bed12 files using bam files with secondary alignments removed, and then performed "flair correct" on those bed12 files to obtain psl files. With psl files, we then performed "flair collapse" on all samples concatenated fastq file to obtain a collapsed fasta file. Lastly, we performed "flair quantify" with the collapsed fasta file and each fastq file again to quantify for each sample. For TALON, we first initialised the database with "talon_inititalize_database" and we then

performed “talon_label_reads” for each sample bam file with a MD tag added. We then performed “talon” to discover novel transcripts and after which, “talon_abundance” to quantify transcript expression for each sample. For LIQA, we first filtered bam files with “samtools -F 2308 -q 50” and then performed “liqa -task quantify” with filtered bam files and “-max-distance 10 -f_weight 1” as recommended in the manual. For NanoCount, we used the recommended “-ax map-ont -p 0 -N 10” to align reads to transcriptome reference with minimap2 (version 2.17) [14] and output TPM values are used in comparison. Note that for NanoCount, we excluded the top three largest samples as we were unable to run NanoCount successfully due to memory issues. For Salmon, we followed the ONT pipeline (https://github.com/nanoporetech/pipeline-transcriptome-de) where we used “-ax map-ont -p 1.0 -N 100” parameters to align reads to transcriptome reference with minimap2 (version 2.17) and then “quant --ont -l U” parameters for salmon. For completeness, we have also included FLAMES [15] and IsoQuant, two other available long read transcript discovery and quantification methods for the quantification benchmark. For FLAMES, we processed fastq files following the usage with bulk data analysis suggestion with example SIRV_config.json. For IsoQuant, we processed bam files with “--data-type nanopore” parameter. For StringTie2, FLAIR, IsoQuant and Salmon, the output transcript per million (TPM) expression levels are used for comparison. For TALON, FLAMES, LIQA and featureCounts, the reported counts per transcript were normalised to counts per million (CPM) for comparison. For NanoCount, the output abundance estimates were used for comparison.

3. Full-length and unique read support

Bambu provides full-length and unique read support estimation for each transcript in each sample. To assess the impact of estimation uncertainty due to missing data, we performed two analyses.

In the first analysis, we applied CPM, unique read count, and full-length read count based filtering on the v0.3 SG-NEx samples with sufficient number of aligned reads (400000, n = 73) to assess the influence of filtering in providing more stable estimation. For samples generated using the same protocol within each cell line, we calculated the intra-sample CPM estimates correlation on filtered transcripts based on average CPM, unique read count, or full-length read count passing varying thresholds from 1 to 20, and we took the average correlation.

To evaluate the efficacy of quantification of tools in the presence of in-active transcripts, we generated artificial spliced-isoforms which could be assigned read support without full-length and unique reads. To avoid overly complex scenarios, we selected genes with average expression levels greater than 100 CPM, a partial read support fraction greater than 30%, less than 8 isoforms, and with no isoforms having 3 or more exons. For the selected genes, we identified the most abundant isoform as the reference isoform on which to base the artificial spliced-isoform. From the internal exons of this isoform, we removed the two most commonly used exons among the other isoforms of this gene generating an artificial exon-skipping event. When there are four or more equally commonly used internal exons, we will randomly choose four to remove to mitigate the complexity in such cases. We run Bambu without discovery on the v0.3 SG-NEx Hct116 samples (n = 13) using the complete annotations that included the artificial transcripts.

To assess the performance of using full-length and unique support to determine if a transcript is truly expressed, we compared CPM, full-length read count, full-length CPM, unique read count, and unique CPM filters at thresholds from 1 to 20. As an additional comparison check, we also ran Salmon and NanoCount using the same artificial annotations, using both SalmonTPM, and NanoCountCPM as filters respectively. For NanoCount, we excluded the two largest samples for this analysis as we were unable to run

them with NanoCount due to memory issues. We measured sensitivity as the percentage of transcripts that pass the thresholds for each filtering method among all annotated transcripts, and the precision is calculated as the percentage of valid (non-artificial) transcripts among all transcripts that pass the thresholds for each filtering method. We averaged the sensitivity and precision across the Hct116 samples.

4. Quantification of retrotransposon-derived isoforms

To identify retrotransposon-derived genes and isoforms, we focused on the v0.3 SG-NEx hESC samples (n = 14) and ran Bambu with a NDR threshold of 0.3 as we expected an enrichment of novel transcripts due to the abundance of repeat elements compared to the other v0.3 SG-NEx cell line samples (n = 51). The extended annotation output from Bambu was then overlapped with the RepeatMasker [16] sequences matching the Grch38 Ensembl annotations release version 91 [10]. To identify the top expressed repeat element types, we ranked the number of novel spliced isoforms expressed with a CPM level greater than or equal to that do not overlap with any canonical annotations in any of the hESC samples for each repeat element type.

To quantify the overlapping percentage of repeat elements for each transcript, we looked at the overlap for each exon within each transcript for each repeat. Exons not overlapping with canonical annotations are noted as novel exons. The average overlap percentages were calculated within annotated or novel exons first and then summed up to provide the overall overlap percentage for the transcript. We then compared the distribution of overlapping percentages between annotated isoforms and novel isoforms. We focused on annotated exons in annotated isoforms and novel exons in novel isoforms for this comparison to reduce the potential biasing towards the null hypothesis. A paired t-test [17] with unequal variance assumption was then conducted to assess the level of significance between the mean

overlapping percentages, with two-sided two-sample Kolmogrov-Smirnov test [18] performed to test the differences in the distributions of the overlapping percentages.

To identify the number of novel genes transcribed from HERVH-LTR7 repeats, we shortlisted novel isoforms, including non-canonical overlapping novel isoforms, re-arranged, or novel spliced canonical isoforms that are overlapping with HERVH-LTR7 repeats, and ranked them by their average expression levels across the hESC samples. We then identified the top expressed isoforms that constitute 90% of the overall isoform expression. To understand the expression of these retro-transposon isoforms in cancer cell lines, we performed Bambu with the extended annotation obtained from hESC samples on cancer cell lines for quantification. Note that for this analysis, HEYA8 samples were not included due to a sample cross-contamination in reads caused by the embedded de-multiplexing protocol used by Guppy (version 3.2.10) [19] during the basecalling process. We also performed Bambu on hESC samples without discovery to understand the impact of not discovering these retro-transposon genes/isoforms.

References

1. Leger, A. NanoCount Manual. https://a-slide.github.io/NanoCount/ (2020).
2. Sipos, B., Rudd, S. & Love, M. pipeline-transcriptome-de/config.yml at master · nanoporetech/pipeline-transcriptome-de. *GitHub* https://github.com/nanoporetech/pipeline-transcriptome-de (2021).
3. Kovaka, S. *et al.* Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *Genome Biol.* **20**, 278 (2019).
4. Wyman, D. *et al.* A technology-agnostic long-read analysis pipeline for transcriptome

discovery and quantification. *bioRxiv* 672931 (2020) doi:10.1101/672931.

5. Tang, A. D. *et al.* Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns. *Nat. Commun.* **11**, 1438 (2020).
6. Prjibelski, A. D. *et al.* Accurate isoform discovery with IsoQuant using long reads. *Nat. Biotechnol.* (2023) doi:10.1038/s41587-022-01565-y.
7. Pertea, G. & Pertea, M. GFF Utilities: GffRead and GffCompare. *F1000Res.* **9**, 304 (2020).
8. Hardwick, S. A. *et al.* Spliced synthetic genes as internal controls in RNA sequencing experiments. *Nat. Methods* **13**, 792–798 (2016).
9. Conesa, A. *et al.* A survey of best practices for RNA-seq data analysis. *Genome Biol.* **17**, 13 (2016).
10. Aken, B. L. *et al.* The Ensembl gene annotation system. *Database* **2016**, (2016).
11. Hu, Y. *et al.* LIQA: long-read isoform quantification and analysis. *Genome Biol.* **22**, 182 (2021).
12. Gleeson, J. *et al.* Accurate expression quantification from nanopore direct RNA sequencing with NanoCount. *Nucleic Acids Res.* **50**, e19–e19 (2021).
13. Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* **14**, 417–419 (2017).
14. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
15. Tian, L. *et al.* Comprehensive characterization of single-cell full-length isoforms in human and mouse with long-read sequencing. *Genome Biol.* **22**, 310 (2021).
16. Smit, A. F. A., Hubley, R. & Green, P. RepeatMasker. Preprint at (1996).
17. Semenick, D. TESTS AND MEASUREMENTS: The T-test. *Strength & Conditioning Journal* **12**, 36 (1990).

18. Massey, F. J. The Kolmogorov-Smirnov Test for Goodness of Fit. *J. Am. Stat. Assoc.* **46**, 68–78 (1951).
19. Wick, R. R., Judd, L. M. & Holt, K. E. Performance of neural network basecalling tools for Oxford Nanopore sequencing. *Genome Biol.* **20**, 129 (2019).