

# *SPARK: in silico* simulations for benchmarking nascent RNA sequencing experiments

Ezequiel Calvo-Roitberg<sup>1</sup>, Jesse W. Lehman<sup>1</sup>, Edric Tam<sup>2</sup>, Shaimae Elhajjajy<sup>4</sup>, Barbara E. Engelhardt<sup>2,3,†</sup>, Athma A. Pai<sup>1,†</sup>

1. RNA Therapeutics Institute, UMass Chan Medical School, Worcester, MA
2. Department of Biomedical Data Science, Stanford University, Stanford, CA
3. Gladstone Institutes, San Francisco, CA
4. Department of Genomics and Computational Biology, UMass Chan Medical School, Worcester, MA

† corresponding authors: [barbara.engelhardt@gladstone.ucsf.edu](mailto:barbara.engelhardt@gladstone.ucsf.edu) and [athma.pai@umassmed.edu](mailto:athma.pai@umassmed.edu)

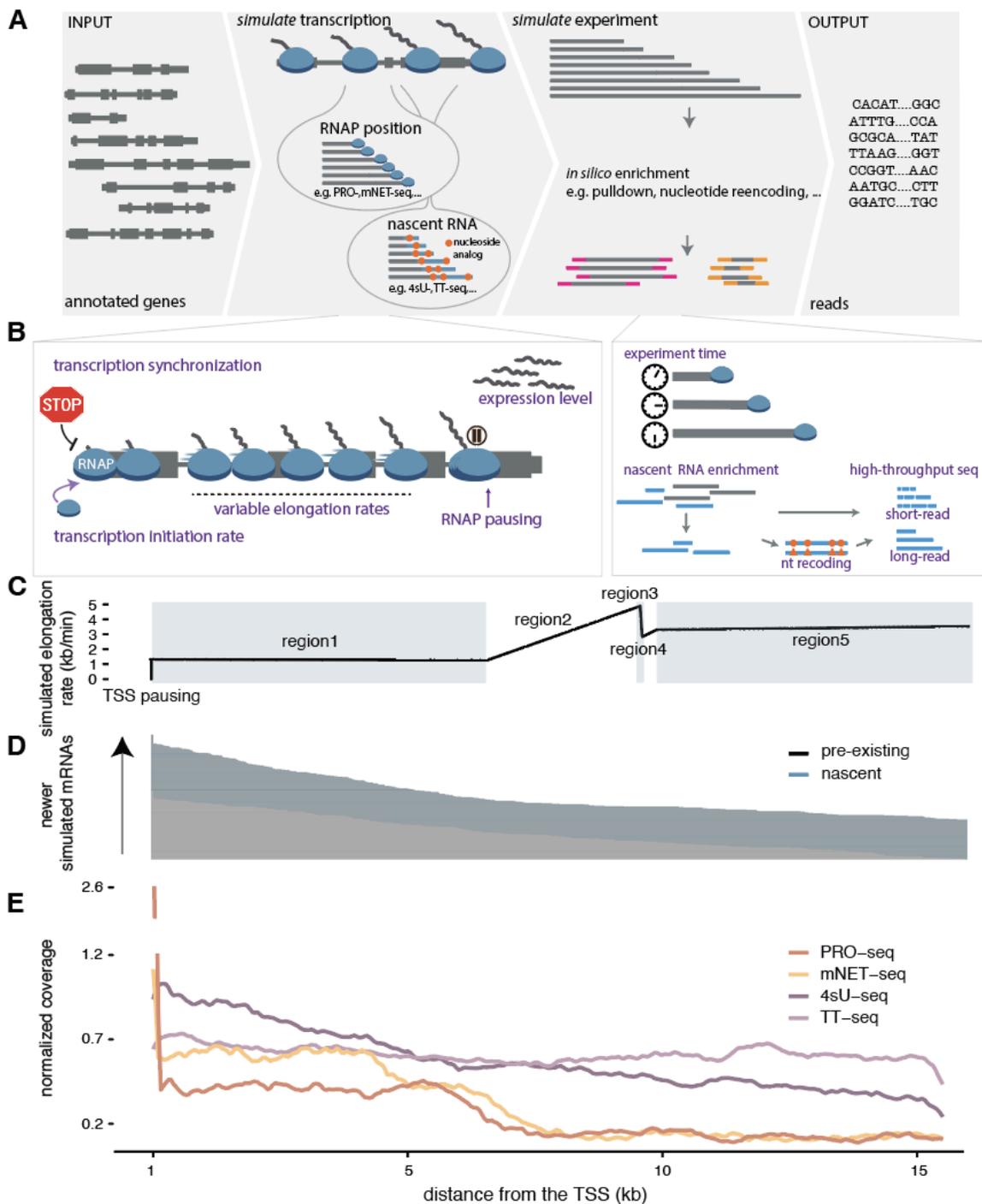
## **ABSTRACT**

Nascent RNA sequencing offers profound insights into transcriptional dynamics, yet there are substantial challenges to analyzing these data. The development of proper computational tools necessitates realistic benchmarking datasets that reflect biological variability and technical biases. We present *simulated pre-mRNA and RNA kinetics (SPARK)*, a versatile *in silico* framework for generating reads across nascent RNA sequencing approaches. SPARK simulates the process of transcription — allowing for variable elongation rates and pausing events — and key experimental features. SPARK provides a comprehensive platform for computational development and benchmarking in nascent RNA genomics.

RNA sequencing (RNA-seq) has transformed transcriptome analysis by enabling precise and accurate gene expression estimation. However, standard RNA-seq approaches only provide a static snapshot of steady-state mature mRNA levels, which limits our ability to understand the gene regulatory mechanisms that give rise to dynamic mRNA profiles across cellular contexts. Instead, it is increasingly clear that directly characterizing nascent RNA levels and transcriptional dynamics may yield greater insights into the mechanistic steps that underlie mRNA biogenesis.

Consistent with this realization, recent advances in the development of genome-wide sequencing approaches to profile ongoing transcription and nascent RNA have reshaped our view of early gene regulation and mRNA biogenesis. These include methods designed to assess the dynamics of RNA polymerase (RNAP) position—such as Precision Run On (PRO)-seq<sup>1</sup> and [mammalian] Native Elongating Transcript (NET)-seq<sup>2,3</sup>—and quantifying the abundance of nascent RNA — such as metabolic labeling with nucleotide analogs (e.g., 4-thio-uridine (4sU)-seq<sup>4</sup> and Transient Transcriptome (TT)-seq<sup>5</sup>), and nucleotide recoding (NR-seq; e.g., SLAM-seq<sup>6</sup> and TimeLapse-seq<sup>7</sup>). Together, these approaches promise the ability to directly profile newly synthesized transcripts to monitor the real-time transcriptional activity with temporal resolution that sequencing mature RNA cannot offer.

The growing variety and adoption of nascent RNA sequencing methods also come with a need for robust computational tools that accurately model transcriptional processes. While nascent RNA-specific analytical tools have become more prevalent<sup>7-10</sup>, there is not yet a robust simulation framework to generate realistic benchmarking datasets and validate methods that quantify transcription rates. Most currently available pipelines to simulate nascent RNA<sup>11,12</sup> generate count matrices across nucleotides or genes, which are only useful for end-stage quantitative model development. Count-based simulations do not capture the full range of biological and technical features inherent to these datatypes (e.g., variability in rate parameters across a gene, nucleotide substitutions, artefacts of library preparation, mapping biases) that must be incorporated into computational frameworks to obtain precise and accurate estimates of transcriptional dynamics.

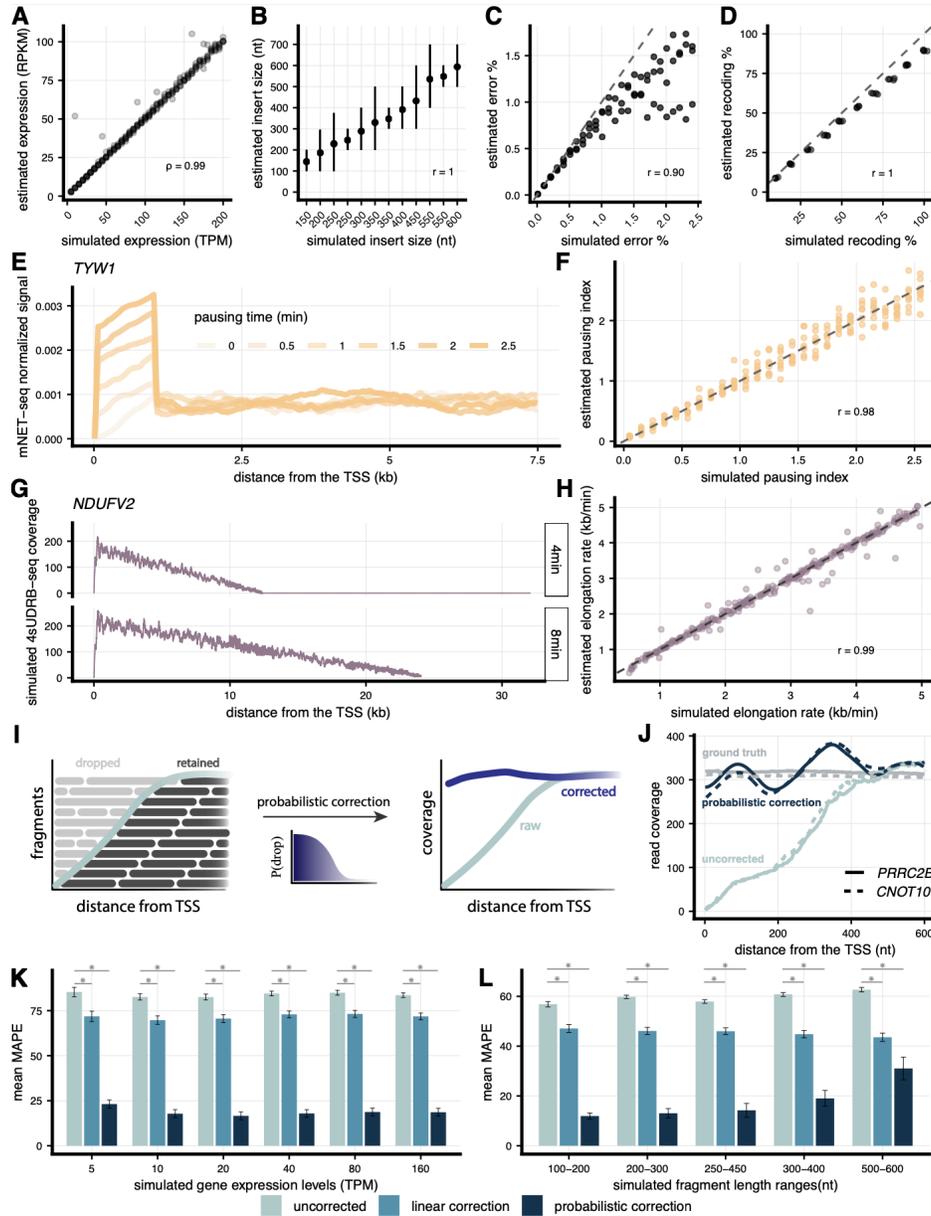


**Figure 1. SPARK simulates reads from nascent RNA sequencing methods.** (A) Schematic overview of SPARK (B) SPARK allows for the modulation of various biological (*left*) and technical (*right*) parameters relevant to nascent RNA experiments. (C) Representative example of simulating nascent RNA reads from *SLC25A19* using SPARK. Elongation rates are varied across the gene, with promoter-proximal pause and 5 distinct elongation regions. (D) Simulated RNAs as horizontal lines, with pre-existing (*black*) and newly-synthesized (nascent; *blue*) regions for each molecule. (E) Simulated normalized read coverage for four nascent RNA sequencing methods, simulated using identical biological parameters.

To overcome these limitations, we introduce *simulated pre-mRNA and RNA kinetics (SPARK)*, a comprehensive simulation framework for generating nascent RNA benchmarking datasets. *SPARK* uses real gene sequences to simulate nascent RNA sequencing reads after performing *in silico* transcription, nascent RNA enrichment, and library preparation steps (**Fig. 1A**). To capture the inherent diversity in gene sequences and architectures that may bias experimental or analytical steps, *SPARK* first clusters genes based on several relevant features and chooses a user-determined number of genes across clusters (**Supplementary Fig. 1A**).

Despite increasing evidence that RNAP elongation rates vary across and within genes due to sequence, chromatin, or other regulatory features<sup>13,14</sup>, most RNA simulators assume constant transcription rates across genes. This forestalls the ability to benchmark computational tools to estimate local elongation rates from RNAP localization or nascent RNA abundance data. *SPARK* allows flexibility to simulate either constant or variable elongation rates across a gene by assigning nucleotide-specific elongation rates across a gene before *in silico* RNA generation (Methods; **Fig. 1B left panel**; **Supplementary Fig. 1B**). Additionally, *SPARK* can vary transcription initiation rates, add promoter-proximal or gene-body RNAP pausing events, and replicate experiments with transcription synchronization with drugs (e.g. 4sUDRB-seq).

Using this backbone, *SPARK* can simulate conditions across a wide variety of nascent RNA sequencing approaches. During the process of *in silico* RNA synthesis, users can choose methods that either mimic metabolic labeling or labeling of RNAP positions. To simulate metabolic labeling approaches like 4sU-seq or TT-seq, *SPARK* simulates the stochastic incorporation of nucleotide analogues (e.g., 4sU, 5-ethynyluridine (5eU)), while the incorporation of a “biotin-NTP” recreates the process of transcription termination inherent to run-on assays like PRO-seq (Fig. 1A, middle left panel). Users define the time available for *in silico* RNA synthesis, which determines how many new RNAP can be initiated and how far simulated engaged RNAP can travel in a gene (**Fig. 1B, right panel**). Following synthesis, *SPARK* enriches for nascent RNA depending on the chosen experimental model, by recreating biotin-NTP or RNAP pulldowns followed by MNase digestion (PRO-seq or mNET-seq, respectively) or biotinylation and streptavidin pulldown of nucleoside analogues (e.g. 4sU-, TT-, or 5eU-seq), before proceeding to either short- or long-read library preparation with varied technical parameters (**Supplementary Fig. 1C,D**). For metabolic labeling modes, users can also select nucleoside recoding to induce substitutions at labeled positions (e.g. T > C substitutions for 4sU).



**Figure 2. SPARK accurately recapitulates data from nascent RNA-seq approaches.** Correlations between simulated parameters (*x*-axis) and estimated parameters from simulated reads (*y*-axis) for **(A)** gene expression levels (Spearman's  $\rho = 0.99$ ), **(B)** insert sizes (Pearson's  $R = 1$ , where points represent mean and vertical bars show full range), **(C)** sequencing error rates (Pearson's  $r = 0.9$ ), and **(D)** T>C nucleotide recoding frequencies from a TT-NR-seq experiment (Pearson's  $r = 1$ ). **(E)** Read coverage for *TYW1* from mNET-seq simulations across different promoter-proximal pause durations. **(F)** Correlation between simulated (*x*-axis) and estimated (*y*-axis) pausing indices from mNET-seq simulations read coverage data (Pearson's  $R = 0.98$ ). **(G)** Read coverage for *NDUFV2* from 4sUDRB-seq simulation. **(H)** Correlation between simulated (*x*-axis) and estimated (*y*-axis) elongation rates from 4sUDRB-seq simulations (Pearson's  $R = 0.99$ ). **(I)** Schematic of probabilistic edge effect correction. **(J)** Ground truth, raw, and corrected read coverage for 2 genes (simulated fragment size = [300-400] and TPM = 50). Symmetric mean absolute percent error (MAPE) for three methods across **(K)** gene expression levels and **(L)** fragment lengths ( $n = 48$  per bin) (adjusted  $p$  value <  $10^{-9}$  for all comparisons).

To illustrate the versatility of *SPARK*, we simulated reads from a single gene using the same biological parameters across four different experimental modes. This simulation included a promoter-proximal pausing event and five regions across which RNAP elongation rates changed (**Fig. 1C**) to generate 5000 RNA molecules (**Fig. 1D**), 44% of which derived from RNAP initiated during the simulated experiment (fully nascent) and with the rest from previously engaged RNAP. Each simulated experiment shows expected read coverage profiles (**Fig. 1E**), with an enrichment of reads at the beginning of the gene for approaches that map RNAP position (PRO-seq and mNET-seq) and more gene-body coverage in metabolic labeling approaches, with reduced 5' bias in TT-seq relative to 4sU-seq<sup>5</sup>. Finally, the read coverage recapitulates simulated biological parameters, with marked RNAP accumulation near the pause site and reduced density in regions of rapid elongation.

A distinguishing feature of the *SPARK* framework is the simulation of many different steps and biases in nascent RNA library preparation protocols (**Supplementary Fig. 1C,D**). To confirm that *SPARK* accurately simulates these technical parameters, we looked at the correlations between ground truth and parameter estimations from simulated reads for four different features: gene expression levels, insert sizes generated from fragmentation during library preparation, sequencing error rates, and efficiency of nucleotide recoding (**Fig. 2A-D**). For all four of these parameters, the estimates from simulated reads closely recapitulate the ground truth (Spearman or Pearson correlations > 0.9 for all parameters). Notably, we see that high rates of random sequencing errors or nucleotide recoding are underestimated in the simulated data, likely due to an increase in erroneous mapping of reads<sup>8</sup>. This bias highlights the need for simulation pipelines that incorporate technical biases and output read sequences rather than counts.

The main purpose of *SPARK* is to generate biological data with known ground truths that can be used to benchmark quantitative models. Thus, we simulated mNET-seq data across a range of promoter-proximal pausing durations. We see that longer pause durations result in increased TSS read coverage, reflecting prolonged RNAP residence (**Fig. 2E**)<sup>2,3</sup>. More broadly, we see that pausing indices estimated from simulated data closely recapitulate the ground truth parameters (**Fig. 2F**; Pearson correlation = 0.98). We next simulated 4sUDRB-seq data, which involves transcription synchronization followed by 4sU labeling and enrichment of nascent RNA after 4 and 8 minutes of re-initiated transcription<sup>15</sup>. Consistent with expectation, simulated data show 5' biased signal that progresses over time as RNAP elongates across the gene (**Fig. 2G**)

and elongation rates estimated from simulated data closely recapitulate ground truth elongation rates (**Fig. 2H**).

Finally, to demonstrate the utility of *SPARK*, we used *SPARK* simulations to test a novel computational method to correct for RNA-seq edge effects. During short-read library preparation, random fragmentation followed by size selection leads to a depletion of fragments—and ultimately sequencing coverage—at the ends of RNA molecules, leaving these regions systematically underrepresented<sup>16</sup>. This limitation is especially problematic when mapping RNAP position and nascent RNA, both of which derive substantial biological insights from 5' regions of mRNA molecules. Using *SPARK* simulated reads to model this effect and focusing on 5' regions of genes, we developed a correction approach that relies on modeling the probability that a fragment overlapping any nucleotide in the 5' end is dropped during size selection (**Fig. 2I-J**, Online Methods). While a linear correction (Methods) improves edge coverage relative to uncorrected coverage (average 13.2% decreased MAPE), our probabilistic correction provides a substantially better improvement across a broad range of gene expression levels (average 65.1% decreased MAPE; **Fig. 2K**,) and fragment length ranges (average 41.7% decreased MAPE; **Fig. 2L**).

*SPARK* is a versatile and comprehensive computational framework designed to generate *in silico* nascent RNA sequencing reads that accurately mimic experimental data, enabling the development, optimization, and validation of nascent RNA analytical workflows. One last feature to highlight is the ability to include background RNA molecules into *SPARK* simulated data, crucial for developing models to accurately quantify and handle noise in nascent RNA datasets. Finally, a key limitation of *SPARK* is the current lack of support for simulating RNA processing (*i.e.* RNA splicing, cleavage, secondary structure, etc.) variability and dynamics, which are likely to hugely shape the biogenesis and stability of nascent RNA molecules. This will be a crucial expansion for future work.

## CODE AVAILABILITY

*SPARK* and the code for downstream analyses included in this manuscript are available at <https://github.com/thepailab/spark>. The edge effect corrector implemented in this manuscript is available at <https://github.com/edrictam/EdgeEffectCorrector>.

## **ACKNOWLEDGEMENTS**

We thank Alexander Park for work on early versions of the simulation scripts and Adam Siepel, Engelhardt lab, and Pai lab for helpful discussions, suggestions, and comments. This work was funded by grants R35GM133762 (AAP), R01HG012944 (AAP), R01HG013736 (BEE), and R01HG012967 (AAP and BEE) from the National Institutes of Health, the Parker Institute for Cancer Immunology (PICI), the Biswas Family Foundation, the Warren Alpert Foundation (ET), the Croucher Foundation (ET) and the Chan-Zuckerberg Institute (BEE). BEE is a CIFAR Fellow in the Multiscale Human Program.

## ONLINE METHODS

### **SPARK framework**

The basic *SPARK* framework involves (1) selection of genes, (2) assignment of elongation rates across a gene, (3) *in silico* simulation of RNA molecules, and (4) simulating enrichment of nascent RNA through one of four experimental modes, with many optional parameters to modify experimental conditions as desired.

Gene clustering and selection. *SPARK* is designed to select genes from which to simulate nascent RNA reads that represent a diversity of genomic features. Users specify the number of genes from which they would like to simulate reads and provide: (1) a gtf with gene annotations and (2) a reference fasta file. For each gene, the longest isoform (upstream-most transcript start site and downstream-most transcript end site) is selected, and its features are used for classification. Genes in the gtf are first categorized by gene length, total transcript length(s), number of exons, mean exon lengths, number of introns, mean intron lengths, first intron length, lengths of 5' and 3' untranslated regions (UTRs), and exonic and intronic nucleotide composition. Hierarchical clustering is used to identify clusters of transcripts with similar characteristics. An equal number of genes is selected from each cluster for downstream simulations, ensuring a balanced and unbiased set of genes reflecting genome diversity. *SPARK* can also simulate genes for any species with annotated genomes and selected genes from a pre-filtered gtf file.

Nucleotide-specific elongation rates. *SPARK* simulates variable transcription elongations by assigning a specific rate to each nucleotide within a gene. The user first defines ranges from which *SPARK* selects the number and length of contiguous genomic regions with different elongation rates within each gene. Nucleotides within each region are assigned a specific rate based on a linear interpolation between the rate at the beginning of the region and the rate at the end of the region, assuming a constant change across the entire region. Users can also specify simpler elongation rate profiles, such as simulating a single, constant elongation rate for all nucleotides within a defined region or simulating a constant rate across the gene.

Finally, users can choose to include RNAP pausing events throughout the gene. Pause sites can be placed randomly across the gene or at specific genomic locations (e.g. the TSS). These events are defined as single-nucleotide regions. The duration of the pause is user-defined and

used to define the time required to traverse that specific nucleotide. Therefore, a pausing event is equivalent to the RNAP traversing a single nucleotide extremely slowly.

Simulation of RNA synthesis. RNA molecules are *in silico* synthesized based on a series of rates that determine the distribution of RNAP positions across a gene at the start, during, and at the end of each simulated experiment. *SPARK* simulates individual initiation events as a Poisson process based on user-specified ranges of transcription initiation rates  $i$ , where the time interval between consecutive initiation events is stochastically drawn from an exponential distribution with mean  $i^{-1}$ . The total time required for an RNAP to traverse the gene is calculated by summing the cumulative elongation times (gene length  $L$  divided by the average elongation rate  $e$ ) and the sum of pausing durations across pause sites  $p$ . This allows for the calculation of the average number of polymerases  $N_{RNAP}$  simultaneously transcribing a gene at steady-state, which is the product of the initiation rate and the traversal time:

$$N_{RNAP} = i \times \left( \frac{L}{e} + p \right)$$

To establish the initial state of the simulation, *SPARK* first populates the gene with  $N_{RNAP}$  RNAP that would be present (assuming steady-state ongoing transcription). The position of each RNAP is assigned probabilistically. First, RNAPs are assigned to genomic regions (e.g., elongation regions or pause sites) probabilistically given the relative time required to traverse that region. Second, RNAPs are assigned to precise nucleotide positions chosen from a uniform distribution across the given region. This weighted assignment of RNAP locations ensures that regions with longer traversal times have more RNAP at steady-state.

*SPARK* simulates a target number of total mRNA molecules  $N_{RNA}$  (default 5,000) by calculating the number of independent gene copies  $N_{gene}$  (analogous to DNA molecules across a population of cells) required to generate that number of molecules during the simulation. This is determined by dividing the target count by the total mRNAs produced per copy, which is the sum of newly initiated polymerases ( $i \times t$ , where  $t$  is the simulation time) and  $N_{RNAP}$ :

$$N_{RNA} = (i \times t + N_{RNAP}) \times N_{gene}$$

Each gene copy is simulated as an independent transcriptional locus.

### **SPARK experimental modes**

Following the synthesis of RNA molecules, *SPARK* can generate reads from one of four nascent RNA experimental approaches: metabolic labeling with full-molecule nascent RNA enrichment, metabolic labeling with fragmented nascent RNA enrichment, RNAP run-on, or nascent RNA associated with elongating RNAPs.

Metabolic labeling and nascent RNA enrichment. *SPARK* simulates metabolic labeling experiments (e.g., 4sU-seq, TT-seq) by first simulating the stochastic incorporation of nucleotide analogs (e.g. 4sU). Users specify the incorporation rate, which defines the probability that an analog will be incorporated instead of the canonical nucleotide (e.g., uridine). The subsequent *in silico* enrichment step is performed differently depending on the specific protocol being simulated (specified below).

Nascent RNA enrichment. The nascent RNA enrichment experimental mode (--experiment\_type nascentrnapt) simulates a 4sU-seq style nascent RNA pulldown experiment by conditioning on full-length molecules that have at least one nucleotide analog incorporated. Following the simulation of full-length RNA molecules, discards imulated molecules that do not have at least one incorporated nucleotide analog. The remaining set of full-length labeled transcripts is then passed to subsequent library preparation steps.

Transient transcriptome (TT)-seq. The TT-seq experimental mode (--experiment\_type ttseq) simulates a TT-seq style experiment by performing *in silico* fragmentation (see below) directly after full-length RNA generation and labeling but before nascent RNA enrichment. After this fragmentation, *SPARK* enriches for nascent RNA fragments (rather than nascent RNA molecules). Fragments that do not contain at least one nucleotide analog are discarded from the population of fragments before proceeding to library preparation.

Precision Nuclear Run-on (PRO)-seq. The PRO-seq experimental mode (--experiment\_type proseq) simulates a run-on assay by simulating the incorporation of a biotinylated nucleotide (biotin-NTP, where N can refer to A|C|G|U and can be user-defined) during the *in silico* transcription process. Users specify an incorporation rate, which defines the probability of a biotin-NTP being incorporated instead of a canonical nucleotide. The most upstream biotin-NTP

incorporation event then defines the 3' terminus of the nascent transcript by truncating the molecule at that nucleotide. Truncated molecules then proceed to the *in silico* fragmentation module (see below), in which only the 3'-most fragment containing the singular biotin-NTP is retained for subsequent library preparation simulation.

mammalian Native Elongating Transcript (mNET)-seq. The mNET-seq experimental mode (--experiment\_type mnetseq) simulates RNAP pulldown-based nascent RNA sequencing assays by identifying all molecules actively undergoing elongation at the end of the simulated experiment time. The simulation mimics RNAP pulldown followed by MNase digestion. For each actively transcribing molecule (RNAP has not reached the 3' end of the gene), *SPARK* extracts the 3'-terminal fragment that represents the nascent RNA physically protected by the RNAPII complex by selecting the last N nt of the molecule, where N is drawn from a uniform distribution between 35-100 nucleotides<sup>2</sup>. Only these 3' fragments are retained for subsequent library preparation steps.

### ***SPARK in silico* library preparation**

The final component of *SPARK* is to perform *in silico* library preparation from the enriched nascent RNAs following simulation of experimental approaches, resulting in high-throughput sequencing reads.

Fragmentation and size selection. For experimental modes that include a random fragmentation step during library preparation (e.g., 4sU-, TT-, and PRO-seq), *SPARK* simulates RNA fragmentation as described before<sup>18,19</sup>. Briefly, the length of the fragments is drawn from a modified Weibull distribution with  $\delta = \log_{10}(length)$  and  $\eta =$  mean of the selected fragmentation length interval. This will sample  $N_{transcripts}$  such that:

$$N_{transcripts} = \frac{mRNAlength}{\eta \Gamma(1 + \frac{1}{\delta})}$$

These breakpoints then give n length that can be transformed into a fragment of length (*di*) by:

$$di = mRNAlength \frac{X_i^{\frac{1}{\delta}}}{\sum_j x_j^{\frac{1}{\delta}}}$$

For experimental modes that involve metabolic labeling (nascent RNA pulldown and TT-seq), only fragments within the specified fragment length range are retained in the final fragment pool. Neither PRO- nor mNET-seq approaches have size selection steps.

Read generation. *SPARK* can generate either short or long high-throughput sequencing reads. For short reads, simulated reads are generated from N nucleotides at the beginning, end, or both (for paired-end sequencing) of the fragments or full-length molecules if no fragmentation was performed), where N is a users-specified read length. The default library orientation simulates a dUTP-based protocol (fr-firststrand). For a paired-end simulation, this results in the first read (R1) being the reverse complement of the RNA strand, and the second read (R2) having the same orientation as the RNA strand. *SPARK* can also simulate fr-secondstrand and unstranded library configurations (Fig. S1B). For the generation of long reads, the entire simulated mRNA molecule is output as a read. Users can specify the strand orientation to emulate different long-read technologies. This includes options for 3' to 5' strandness (mimicking direct RNA sequencing) or unstranded reads (5' to 3' or reverse complement 3' to 5', mimicking cDNA sequencing), as seen in Oxford Nanopore library preparation methods.

Sequencing Errors. *SPARK* can simulate stochastic sequencing errors. A user-specified rate dictates the probability that any given nucleotide in a simulated read will be substituted with a different, randomly chosen nucleotide. The default error rate is 0.01%, reflecting the typical error profile of Illumina sequencing platforms. For long-read sequencing, this error rate should likely be raised to 5-10%.

### ***SPARK* optional experimental conditions**

Background molecules. Since most genomic approaches involving either biotin or antibody-based pulldowns contain some level of background pulldown, *SPARK* allows users to include background molecules from the simulated genes by specifying the proportion of contaminating molecules coming from noisy pulldowns. These background molecules are simulated as full-length RNA molecules with coverage across the whole gene.

Nucleotide recoding. The usage of metabolic labeling with synthetic nucleotide analogs allows for nucleoside recoding, in which substitutions are introduced during reverse transcription and library preparation (e.g., T>C or G>A for 4-thiouridine and 6-thioguanine, respectively) at locations of an incorporated analog. To simulate nucleotide recoding, *SPARK* allows users to

specify the desired substitutions and a recoding rate, which defines the probability that a substitution will be introduced at any given position where a nucleotide analog has been incorporated (which is conditional on the incorporation rate above).

RNA synchronization. *SPARK* can simulate the synchronization of RNAP transcription, in which drugs (e.g. DRB) are used to block new initiation or elongation for a time period that allows ongoing elongation to complete RNA synthesis before allowing new transcriptional cycles to start. This is a common experimental strategy (e.g., 4sUDRB-seq) to measure elongation rates. In this mode (--drb), the simulation does not generate a steady-state population of pre-existing, elongating polymerases. Instead, all simulated nascent transcripts originate from RNAP molecules that initiate transcription after the simulated experimental start time, mimicking a synchronized release from the TSS. The initiation of these new transcripts on each gene copy still follows the stochastic Poisson process described previously.

### **Parameters used for test simulations**

For this manuscript, all simulated datasets were generated using a strandedness setting that emulates dUTP-based library preparation (fr-firststrand), simulated read lengths of 2x100nt reads, and library insert sizes of 200-300, unless specified otherwise. Human genes were selected using the v95 gtf file for hg38 reference genome. Simulated reads were aligned to the hg38v95 reference genome using STAR<sup>20</sup> using default parameters unless otherwise specified.

Gene expression estimation. Simulations were performed for ten genes using the nascent RNA pulldown mode across ranges of gene expression levels. Following read mapping, gene-level read counts were obtained using bedtools coverage<sup>21</sup>. Counts were then used to calculate reads Per Kilobase of transcript per Million mapped reads (RPKM), using a total library sequencing depth of 20 million reads.

Fragment size estimation. Simulations were performed for 5 genes using the nascentRNA pulldown mode across a range of insert size distributions. Following read mapping, insert size distributions for each library were estimated using the CollectInsertSizeMetrics command from Picard<sup>22</sup>.

Sequencing error estimation. Simulations were performed for 5 genes using the nascentRNA pulldown mode with single-end 100nt libraries across a range of sequencing error rates. Simulated reads were aligned to the XX reference genome using STAR<sup>20</sup> with the following parameters: `--outSJfilterOverhangMin 20 12 12 12, --outSJfilterCountUniqueMin 2 1 1 1, --alignIntronMin 25, and --alignIntronMax 250000`. Per-base coverage was obtained using `samtools mpileup` run on uniquely mapped reads<sup>23</sup>. The sequencing error rate was estimated by considering the total number of detected nucleotide substitutions reported by `mpileup`, divided by the total number of mapped reads per base.

Nucleotide substitution estimation. Simulations were performed for 5 genes using the TT-seq mode and nucleotide recoding with T>C substitutions across different T>C substitution probabilities and a fixed nucleotide analog incorporation rate of 3%. Following mapping, base-specific recoding was estimated using `samtools mpileup`<sup>23</sup>. T>C substitution frequencies were calculated as the total number of T>C substitutions detected by `mpileup` per position divided by the ground truth number of incorporated nucleotide analogs per base, which is recorded in the simulated read names from *SPARK*.

#### **4sUDRB-seq simulation and elongation rate estimation**

Simulations were performed for 500 genes (all longer than 25 kb) using the nascent pulldown mode and RNAP synchronization at the TSS, followed by nascent RNA *in silico* labeling for 4 or 8 minutes. Simulations used a fixed elongation rate across the entire gene that was varied across genes. After mapping, base-specific coverage was obtained with `bedtools coverage`<sup>21</sup>. To estimate the elongation rate from simulated data, the position of the elongating RNAP wavefront (boundary) was identified for each gene at both time points, as previously described<sup>15</sup>. First, a rough boundary estimate was defined as the first bin (at least 2.5 kb downstream of the TSS) exhibiting zero coverage. A cubic spline was then fit to the coverage data to smooth the signal, and the precise boundary was identified as the most downstream position where the derivative of the spline between adjacent bins approached zero. A linear model was then fit to these boundary positions as a function of time, and the slope of this fit was considered to be the estimated elongation rate.

#### **Pausing index calculation**

Simulations were performed for 10 genes using the mNET-seq mode across a range of TSS pausing durations. After mapping, read coverages for the TSS (TSS to TSS+100nt) and early

gene body (TSS + 250nt to TSS+2,250nt) regions were quantified using bedtools coverage<sup>21</sup>. A pausing index (PI) was calculated from the simulations as the ratio of coverage in the TSS region to the coverage in the gene body region. The ground truth pausing index was calculated as the ratio of the time required for a simulated RNAP to traverse the TSS region relative to the time required to traverse the early gene body (TSS+250 nt to TSS+2250 nt).

### Edge effect correction

We use *SPARK* to generate a dataset with which we could develop a new method to correct for edge effects created during short-read library preparation size selection steps. Simulations were performed for 50 human genes across gene expression levels or fragment size ranges using the nascent RNA pulldown mode. All other biological and technical parameters were held constant. After mapping, base-specific coverage was obtained with bedtools coverage. Though edge effects occur at both ends of RNA molecules, we focused our analyses on 5' regions most affected by this edge effect. To do so, we consider the region from the transcription start site to position  $x_{end} = 2t_{min}$ , where  $t_{min}$  denotes the minimum fragment length selected during size selection. This window size proves to be sufficient to capture the full extent of the 5' edge effect in our simulations.

We developed a probabilistic correction for the coverage data at the previously defined 5' region that models the probability of a given nucleotide being affected by the edge effect. Rather than exactly replicating the simulator's internal fragmentation mechanics, we use approximations that simplify the math and yield a tractable correction. The core idea is to approximate a "dropping probability"  $P_{drop}(x)$  that captures the probability of a read being dropped at any position  $x$ .

Using this quantity, we can compute a corrected coverage level:

$$y_{corrected} = y_{observed} + P_{drop}(x) * y_{target}$$

where  $y_{observed}$  denotes the observed coverage (smoothed using a Savitsky-Golay filter of order 3) and  $y_{target}$  denotes an estimated level of coverage that would have been achieved if no size selection had been performed. Empirically, we set:

$$y_{target} = C \times \max_{i \in [0, x_{end}]} y_{observed}(i)$$

Here, we use the maximum observed coverage in the defined region as a conservative proxy for the target coverage, with  $C$  being a user-chosen scaling constant that allows for adjustments for global coverage level differences. Throughout our analyses, we set  $C = 0.75$ .

Due to constraints from the transcript boundary, the upstream-most fragment of the transcript has a different length distribution than interior fragments, which must be incorporated into the approximation of  $P_{drop}(x)$ . Since we are primarily interested in the effect of dropping short fragments at this edge, we assume that the internal fragment lengths follow a Weibull distribution with scale parameter  $t_{min}$  denoting the minimum fragment length after size selection and shape parameter  $\log_{10}(L)$  where  $L$  is the length of the longest transcript observed. By symmetry, we assume the upstream-most fragment length  $d_1$  also follows a Weibull distribution with identical shape but half the scale. Using this framework, the probability that a position  $x$  is included in the upstream-most fragment and dropped is given by:

$$P(x \leq d_1 \leq t_{min})$$

The probability expression can be calculated as  $F_{d_1}(t_{min}) - F_{d_1}(x)$ , where  $F_{d_1}$  denotes the cumulative distribution function of  $d_1$ . Since the scale parameter that governs  $d_1$  is halved, the probability of  $d_1$  exceeding the maximum fragment length cutoff  $t_{max}$  is empirically very close to zero and not included in the correction above.

Coverage depletion in the 5' edge region does not arise solely from the upstream-most fragment. We therefore also consider the effect of the adjacent fragment immediately downstream (the second upstream-most fragment). We are interested in the probability:

$$P(d_1 < x < d_1 + d_2 \text{ and } (d_2 < t_{min} \text{ or } d_2 > t_{max}))$$

which denotes the probability that the position  $x$  is covered by a second fragment of length  $d_2$  that is dropped. Since  $d_2$  is the length of an internal fragment, it is assumed to be independent of  $d_1$  and follows the same Weibull distribution that governs the length of internal fragments. Since the distributions of both  $d_1$  and  $d_2$  are defined exactly, this probability can be accurately and efficiently evaluated in Python using a standard numerical integration/quadrature function from the library SciPy. We can therefore approximate  $P_{drop}(x)$  as:

$$P_{drop}(x) \approx P(x \leq d_1 < t_{min}) + P(d_1 < x < d_1 + d_2 \text{ and } (d_2 < t_{min} \text{ or } d_2 > t_{max}))$$

This quantity can then be used to estimate  $y_{corrected}$ .

To evaluate the efficacy of our probabilistic correction, we compared it against both the uncorrected raw data and a baseline linear corrector. The linear corrector is computed by fitting a simple linear regression model against coverage data in the region from the transcription start site to  $2 \times x_{end}$ . We compared the coverage levels yielded from these three approaches (raw, linear, and the probabilistic corrector) against the ground truth coverage using the symmetric mean absolute percentage error (symmetric MAPE). More precisely, given the ground truth coverage level  $y_{truth}$  and the corrected coverage level  $y_{corrected}$ , the symmetric MAPE is given by:

$$\frac{2}{x_{end}} \sum_{i=1}^{x_{end}} \frac{|y_{corrected}(i) - y_{truth}(i)|}{|y_{truth}(i)| + |y_{corrected}(i)|}$$

For each fragment size or gene expression level and correction method, we computed the symmetric MAPE for all 50 genes and performed pairwise t-tests to gauge the statistical significance of any performance differences after Bonferroni correction for multiple-hypothesis testing.

## REFERENCES

1. Kwak, H., Fuda, N. J., Core, L. J. & Lis, J. T. Precise maps of RNA polymerase reveal how promoters direct initiation and pausing. *Science* **339**, 950–953 (2013).
2. Nojima, T. *et al.* Mammalian NET-Seq Reveals Genome-wide Nascent Transcription Coupled to RNA Processing. *Cell* **161**, 526–540 (2015).
3. Mayer, A. *et al.* Native elongating transcript sequencing reveals human transcriptional activity at nucleotide resolution. *Cell* **161**, 541–554 (2015).
4. Windhager, L. *et al.* Ultrashort and progressive 4sU-tagging reveals key characteristics of RNA processing at nucleotide resolution. *Genome Res* **22**, 2031–2042 (2012).
5. Schwalb, B. *et al.* TT-seq maps the human transient transcriptome. *Science* **352**, 1225–1228 (2016).
6. Herzog, V. A. *et al.* Thiol-linked alkylation of RNA to assess expression dynamics. *Nat. Methods* **14**, 1198–1204 (2017).
7. Schofield, J. A., Duffy, E. E., Kiefer, L., Sullivan, M. C. & Simon, M. D. TimeLapse-seq: Adding a temporal dimension to RNA sequencing through nucleoside recoding. *Nat. Methods* **15**, 221–225 (2018).
8. Jürges, C., Dölken, L. & Erhard, F. Dissecting newly transcribed and old RNA using GRAND-SLAM. *Bioinformatics* **34**, i218–i226 (2018).
9. Rummel, T., Sakellaridi, L. & Erhard, F. grandR: a comprehensive package for nucleotide conversion RNA-seq data analysis. *Nat Commun* **14**, 3559 (2023).
10. Vock, I. W. *et al.* Expanding and improving analyses of nucleotide recoding RNA-seq experiments with the EZbakR suite. *PLoS Comput Biol* **21**, e1013179 (2025).
11. Zhao, Y., Liu, L., Hassett, R. & Siepel, A. Model-based characterization of the equilibrium dynamics of transcription initiation and promoter-proximal pausing in human cells. *Nucleic Acids Res* **51**, e106–e106 (2023).

12. Zhao, Y. *et al.* Deconvolution of expression for nascent RNA-sequencing data (DENR) highlights pre-RNA isoform diversity in human cells. *Bioinformatics* **37**, 4727–4736 (2021).
13. Jonkers, I., Kwak, H. & Lis, J. T. Genome-wide dynamics of Pol II elongation and its interplay with promoter proximal pausing, chromatin, and exons. *Elife* **3**, e02407 (2014).
14. Calvo-Roitberg, E. *et al.* mRNA initiation and termination are spatially coordinated. *Science* **390**, eado8279 (2025).
15. Fuchs, G. *et al.* 4sUDRB-seq: measuring genomewide transcriptional elongation rates and initiation frequencies within cells. *Genome Biol* **15**, R69 (2014).
16. Carralot, J.-P. *et al.* A novel specific edge effect correction method for RNA interference screenings. *Bioinformatics* **28**, 261–268 (2012).
17. McAdams, H. H. & Arkin, A. Stochastic mechanisms in gene expression. *Proceedings of the National Academy of Sciences* **94**, 814–819 (1997).
18. Pai, A. A. *et al.* The kinetics of pre-mRNA splicing in the *Drosophila* genome and the influence of gene architecture. (2017) doi:10.7554/eLife.32537.
19. Griebel, T. *et al.* Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Res* **40**, 10073–10083 (2012).
20. Dobin, A. *et al.* Sequence analysis STAR: ultrafast universal RNA-seq aligner. **29**, 15–21 (2013).
21. Quinlan, A. R. & Hall, I. M. BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**, 841–842 (2010).
22. Website.
23. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).