A Probabilistic Mixture Model Framework for scRNA-seq Read Simulation

WENSHAN LI[1,2,*], WENBO FANG[1], AO LIU[1], BEIBEI LI[1], JUNJIANG HE[1], HONGXIA WANG[1]
[1]School of Cyber Science and Engineering,
Sichuan University,
No. 24 South Section 1, Yihuan Road, Chengdu,
CHINA

[2]School of Cyber Science and Engineering,
Chengdu University of Information Technology,
No. 24 Section 1, Xuefu Road, Southwest Airport Economic Development Zone, Chengdu,
CHINA

**Corresponding Author*

Abstract: - Single-cell RNA sequencing (scRNA-seq) technologies have provided unprecedented insights into gene expression at the cellular level. Drop-seq is one of the most widely used scRNA-seq protocols, and the rapid development of analytical tools for Drop-seq data has followed. These methods are typically evaluated using spike-in experiments or simulated datasets, as the real-world differential gene expression is often unknown. However, spike-in experiments can be both costly and time-consuming, making simulated datasets a more practical alternative. Despite this, most existing RNA-seq simulators are designed for bulk RNA sequencing, highlighting the need for a specialized scRNA-seq simulating method tailored to Drop-seq technology. In this paper, we present Ds-Sim (Drop-seq reads Simulator), a mixture model-based RNA read simulator that generates sequencing reads mimicking those produced in Drop-seq experiments. Our proposed approach is capable of simulating large-scale Drop-seq reads based on user-defined experimental settings, and the generated data closely approximates real Drop-seq results.

Key-Words: - Single-cell RNA sequencing (scRNA-seq), RNA-seq data simulator, Drop-seq data, Transcript expression, Positional bias modeling, Read alignment.

Received: May 26, 2024. Revised: January 6, 2025. Accepted: March 4, 2025. Published: May 12, 2025.

1 Introduction

Single-cell RNA sequencing (scRNA-seq) has revolutionized cellular genomics by enabling transcriptional profiling at unprecedented single-cell resolution, effectively resolving expression heterogeneity across diverse cell subpopulations. This technological paradigm shift overcomes the critical limitation of conventional bulk RNA-seq approaches, which generate population-averaged measurements that inherently mask cell-to-cell variability, [1]. For example, in tumors, bulk RNA-seq averages signals from cancer cells, immune cells, and stromal cells, preventing the identification of transcriptionally distinct subpopulations, [2]. Similarly, in the immune system, bulk profiling of tumor-infiltrating lymphocytes fails to distinguish between exhausted and effector T cells [3], while in developmental contexts, it obscures the dynamic changes occurring during stem cell differentiation, [4]. These limitations have been overcome by scRNA-seq, which resolves gene expression at the individual cell level, uncovering previously hidden biological complexity. Specifically, scRNA-seq is preferred over bulk RNA-seq in these scenarios, where it reveals distinct subpopulations of cancer and immune cells or identifies rare immune subsets such as exhausted T cells. Moreover, it effectively traces dynamic gene expression during stem cell differentiation and developmental processes, which bulk RNA-seq is unable to capture.

RNA-seq methodologies bifurcate into two principal architectures: (i) full-length category, exemplified by Smart-seq [5], and (ii) tag-based category, typified by Drop-seq [6]. In the latter category, mRNAs are captured by a 30-nucleotide oligo(dT) sequence attached to primer beads, which bind to the poly(A)-tails of mRNAs. This process, known as priming, initiates reverse transcription to

generate complementary DNA (cDNA). Primer beads are tiny beads coated with oligonucleotides that contain a cell barcode and a unique molecular identifier (UMI), allowing transcripts from individual cells to be tracked after sequencing, [7]. The efficiency of this priming process is crucial for accurate gene expression analysis. Primer beads tend to bind to regions that contain a continuous sequence of adenine nucleotides. This may involve the poly(A)-tail participating in the priming process and internal poly(A) sequences. As a conclusion, tag-based methods often produce more reads near the tagged end of a transcript (i.e., the poly(A)-tail), [8]. Since tag-based methods sequence only a portion of each RNA molecule, they generate non-uniform read coverage across the transcript, [9]. In contrast, full-length methods divide the transcript into multiple fragments and sequence them, leading to more uniform read coverage. Despite this difference, tag-based protocols demonstrate superior biological fidelity through digital molecular counting - a capability rooted in their unique molecular tracing architecture. By unambiguously associating each sequencing read with its originating transcript molecule via UMIs, these methods effectively mitigate PCR amplification artifacts that plague full-length protocols. This molecular resolution advantage enables precise transcript enumeration, rendering tag-based approaches particularly advantageous for cell-to-cell comparison studies requiring high quantitative accuracy, [9].

Drop-seq has emerged as the industry-standard tag-based scRNA-seq method, predominantly adopted for its unparalleled capacity to interrogate cellular heterogeneity at scale. This bead-based barcoding system achieves cost-efficient population-scale analyses through optimized molecular capture efficiency and moderate sequencing coverage requirements, [6]. It is particularly advantageous in applications such as analyzing tumor heterogeneity, where it captures diverse cell types within the tumor microenvironment at a large scale [6], and profiling immune responses, where it facilitates the identification of rare immune subpopulations, [7]. Additionally, Drop-seq is effective in developmental biology studies, enabling the reconstruction of dynamic gene expression trajectories during stem cell differentiation. Compared to other scRNA-seq methods, Drop-seq offers higher throughput and lower per-cell costs, making it well-suited for applications requiring large-scale single-cell profiling. According to [10], Drop-seq outperforms several popular single-cell sequencing techniques in terms of both time and cost efficiency. As Drop-seq gains popularity, more computational tools tailored for Drop-seq data analysis have been developed. Those tools are typically evaluated either by spike-in methods or simulation data. However, due to the high cost of spike-in approaches, simulation data becomes a favorable alternative. Unfortunately, most current RNA-seq simulators are designed for bulk RNA-seq, underscoring the necessity for a simulator tailored for single-cell sequencing techniques.

Several recent studies have highlighted the importance of simulating realistic read distributions for accurate analysis of RNA-seq data. [11] reviewed various machine learning approaches for analyzing genomic data and emphasized the need for accurate simulation models to validate downstream analysis pipelines. Similarly, [12] demonstrated how RNA knowledge graph analysis can provide insights into relationships between RNA sequences, which could further improve transcript distribution models in scRNA-seq studies. Additionally, [13] presented a cDNA expression analysis study that highlights the importance of accurate cDNA data in downstream analyses, reinforcing the need for improved simulation methods in scRNA-seq technologies.

Most existing RNA-seq simulation tools predominantly focus on bulk sequencing paradigms, rather than scRNA-seq. Popular bulk RNA-seq simulators such as ART [14], Flowsim [15], Grinder [16], FASTQsim [17], and Polyester [18] successfully approximate real data in bulk sequencing. However, these simulators operate under two limiting assumptions: (i) positionally homogeneous read distribution across transcripts, or (ii) GC-content mediated bias inherited from PCR amplification artifacts [19]. These premises fundamentally misrepresent tag-based scRNA-seq platforms like Drop-seq, where 3'-end capture bias creates non-uniform read distributions. Furthermore, the nascent state of scRNA-seq simulation research has confined most tools to count matrix generation rather than generating sequencing reads. Tools such as Splatter [20], Lun [21], Lun 2 [22], and BASiCS [23] successfully obtain the variable gene expression and dropout rates characteristic of scRNA-seq. Dropout rates here refer to the phenomenon where transcripts that are truly expressed in a cell fail to be detected during sequencing, often due to technical inefficiencies or stochastic capture of mRNA molecules. While these tools capture gene expression trends effectively, they cannot simulate sequencing reads, highlighting the need for a read simulator specifically designed for Drop-seq technology: one capable of generating

mRNA reads that capture the unique distribution shape observed in scRNA-seq data, including the biases introduced by the priming process.

In this paper, we present Ds-Sim (Drop-seq reads Simulator), a mixture model-based RNA read simulator that simulates the sequencing reads of a Drop-seq experiment. In Drop-seq data, reads are not uniformly drawn from a transcript but are more likely to be generated from positions near a poly(A)-region (a region that contains a consecutive sequence of adenine nucleotides). The proposed Ds-Sim learns a poly(A)-bias model from which we can sample the start position for each read. Through this process, we can accurately model the bias introduced by Drop-seq (or other tag-based scRNA-seq methods). Next, we use Polyester [18], a popular bulk RNA-seq simulator, to generate reads from the corresponding positions and introduce potential sequencing errors. Finally, to distinguish the origin of the simulated reads, we generate the barcode (including the 12- nucleotide cell barcode and 8- nucleotide UMI) for each read accordingly. The proposed Ds-Sim serves as an effective simulator for Drop-seq technology. The reads generated by Ds-Sim demonstrate similar characteristics as real Drop-seq data. Extensive experiments demonstrate that Ds-Sim can produce effective simulations under different circumstances. Researchers can use this approach to generate simulated data using scRNA-seq techniques efficiently, eliminating the need for complex and costly experiments.

The main contributions of this work are listed as follows:

1. **Poly(A)-Bias Modeling**: We developed a probability model that captures the poly(A)-bias by estimating read sampling probabilities relative to poly(A)-regions. The model accounts for poly(A) tails at the 3' end as well as internal poly(A)-regions within transcripts.
2. **Read Simulation with Position-Specific Probability Distributions**: By generating a transcript-specific probability distribution based on poly(A) distances and sampling from this distribution, we accurately model the skewed read distribution characteristic of Drop-seq data.
3. **Realistic Read Fragmentation and Sequencing Simulation**: We used the Polyester package to simulate transcript fragmentation and read sequencing, incorporating empirical sequencing error models from Illumina platforms.
4. **Accurate Barcode Generation**: We simulated realistic 12-nt cell barcodes and 8-nt UMIs to label transcripts, ensuring that the simulated data accurately mimics real Drop-seq output.

The remainder of this paper is organized as follows: The second Section introduces the framework of our method and explains the simulator workflow in detail. Section 3 describes the experimental datasets used for evaluation, the steps for data preprocessing, and the evaluation metrics. Section 4 presents and discusses the experimental results. Section 5 concludes the paper and explores potential directions for future work.

2 Method

We propose Ds-Sim, an RNA read simulator for the scRNA-seq technology. Figure 1 (Appendix) illustrates the overall framework of Ds-Sim. To begin with, Ds-Sim takes in two input files. The first one is the transcript count matrix M_{TC}. Each entry of this matrix represents the transcript count of the corresponding cell. M_{TC} is determined by the user. It can also be generated by a count matrix generation tool, which was mentioned in the previous section. The second input file contains the reference transcript sequences, which is usually a FASTA file. Because of the poly(A)-bias that occurs in tag-based scRNA-seq techniques, the read coverage of most transcripts demonstrates a skewed shape. In order to accurately model the read distribution, a poly(A)-bias model is learned from real Drop-seq datasets. Based on this model, the start positions of sequencing reads are obtained for each transcript. Polyester, a widely used conventional RNA read simulator, is then applied to produce the reads according to the input reference transcripts and the corresponding start locations. Finally, for each read, the cell barcode and molecular barcode are added to produce the paired-end sequencing data.

As depicted in Figure 1 (Appendix), the proposed method begins by taking two files: a transcript reference file and a transcript count matrix M_{TC}. Specifically, M_{TC} is structured with each row representing a transcript, each column representing a cell, and each entry indicating the corresponding transcript count. The proposed simulator Ds-Sim proceeds in the following steps:

1. **Read Position Generation**: A poly(A)-bias model is created from real Drop-seq data. For each transcript, Ds-Sim obtains the start positions for sequencing reads based on the trained model.
2. **Fragmentation and Sequencing**: The reference transcripts are fragmented, and sequencing reads are produced starting from the sampled

positions. Every read contains cDNA sequences derived from the input transcripts.

3. **Barcode Generation**: Ds-Sim creates the unique cell barcodes and UMIs for the produced reads by combining random nucleotides.

The final output of Ds-Sim is formatted as FASTA files, which contain the simulated reads based on the input data.

2.1 Read Position Generation

As we mentioned in the previous section, reads produced by Drop-seq are affected by the bias introduced from the poly(A)-regions in a transcript due to the priming process. Compared to other regions of the transcript, a higher proportion of reads originate near poly(A)-regions. As a result, the reads are not uniformly distributed. In this paper, we define a poly(A)-region (PAR) and the corresponding poly(A)-bias (PAB) as follows.

Definition 1 (Poly(A)-region, PAR). A PAR is a region that contains a consecutive sequence of n adenine nucleotides, where $n \geq 20$.

Definition 2 (Poly(A)-bias, PAB). PAB occurs when reads are disproportionately generated from positions near a PAR, which can either be the poly(A)-tail at the 3' end of the transcript or an internal poly(A) sequence.

To account for PAB in the simulation process, we train a probability model on real Drop-seq data (described in Section 3) to capture unique features and the biased shape of the read distributions. The PAB model defines the probability of producing a read at a given location in relation to a PAR. Based on the PAB model, we can obtain a probability distribution for each transcript that specifies the likelihood of initiating a read at different positions.

The start positions of sequencing reads are obtained from the PAB model, ensuring that the positions chosen reflect the inherent poly(A)-bias. The selected positions from this step serve as the starting points for simulated reads.

2.1.1 PAB Modelling

As we mentioned earlier, for Drop-seq and other tag-based methods, the read distribution for each transcript is not uniform. Reads are more likely to originate from the positions near a PAR, which results in a skewed PAB distribution. To model this bias, we propose a PAB model. The proposed PAB model describes the read distribution of Drop-seq data. Based on the PAB model, we assign the probability of producing a read relative to the distance to a PAR.

In this paper, we adopt a mixture model to describe the PAB in the process. The mixture model is composed of two components. To be more specific, the first one is learned from real Drop-seq data, which comes from the empirical distribution of reads. It specifically models the bias introduced by the PARs in a transcript, and also captures the potential biases from a real Drop-seq experiment. The second component is determined by a length factor, which adopts another strategy to generate simulated reads. The PAB model is shown in Equation (1), which specifies the probability of producing a read at position i, the distance to the nearest PAR in the transcript.

$$PAB(x = i) = (1 - \alpha)r_i + \alpha\frac{1}{L} \quad (1)$$

In Equation (1), r_i is the factor learned from real Drop-seq data. The proposed PAB is trained on real Drop-seq reads. Through the alignment process, we can generate the read distribution in relation to the PAR for each transcript, which is described in the following definition.

Definition 3 (Transcript read distribution, TRD). For a given transcript, the transcript read distribution describes the number of reads at the current position in relation to the nearest PAR.

Ultimately, a TRD is generated for each transcript in the training dataset. From the learned transcript read distribution, we can calculate r_i according to Equation (2).

$$r_i = \frac{1}{N}\sum_{j=0}^{N}\frac{x_{ij}}{\max(x_j)} \quad (2)$$

To be more specific, x_{ij} denotes the number of reads at position i, which we can retrieve from the TRD of transcript j. It is divided by the maximum number of reads from the corresponding TRD. N here represents the total number of transcripts in the training data. Since reads may be mapped to incorrect positions during the alignment process, normalizing the number of reads for each transcript mitigates the risk of skewing the overall average due to transcripts with a large number of misaligned reads.

Thereby, r_i estimates the probability of generating a read at the given position in relation to its distance to the nearest PAR. Due to the fact that the RNA-seq process initiates from the 3' end of a transcript, only PARs on the 3' end side will be included in the model. We should notice that if a

transcript does not contain any internal PAR, the only contributing factor should be the poly(A)-tail at the 3' end. In this case, i is equivalent to its distance to the 3' end. Since r_i is calculated from the TRDs, which we build from real Drop-seq data, it can capture the PAB and other potential biases from Drop-seq experiments as well.

Another contributing component in the PAB model is the length factor L. L is determined by the read length in the sequencing process, which is defined as the following definition.

Definition 4 (The read length *R*). Read length in scRNA-seq refers to the number of nucleotides sequenced from each cDNA fragment generated from mRNA transcripts. It determines the amount of sequence information obtained per read, influencing both the genomic mapping accuracy through unique read alignment capacity and expression quantification reliability via transcript coverage depth.

The read length serves as another factor influencing the sampling probability. It is not fixed in real scRNA-seq experiments, and can be user-defined in the simulation procedure. Previous studies have demonstrated that the read length R in scRNA-seq data impacts the TRD for some transcripts, [24]. Specifically, if the length of a transcript is short (or close to R), the TRD of the transcript is likely to be uniform, rather than exhibiting the typical PAB we mentioned before. In other words, the reads tend to be uniformly distributed near a PAR. As a result, the proposed PAB model applies a uniform sampling strategy for the first L nucleotides starting from the PAR. In this paper, we set $L = 4R$. The choice is based on cross-validation results, but users can adjust it according to their own settings.

Furthermore, in the PAB model, a weight factor α is applied to determine the influence of the two components, as described in Equation (3).

$$\alpha(i) = \begin{cases} 1 & if\ i \le L \\ \frac{1}{i} & else \end{cases} \quad (3)$$

To summarize, the PAB model is adjustable by the weight factor α. α is a function of the distance i; it adjusts the contribution of the two components: r_i and L. To be more specific, the PAB model applies a uniform sampling strategy when it is close to a PAR, where the probability of sampling at any given distance is the same. In this case, L is the only contributing component to the model. However, when the value of i becomes larger, the situation becomes different. When i is larger than the length factor L, the other component r_i, which is derived from the TRDs, takes over. In the meantime, the PAB becomes more important to the model. As i grows, the second component L is weighted less, while the first component r_i starts to be dominant. When the distance is large, the PAB model is almost entirely determined by the first component. In the next step, we will generate start positions for the simulated reads based on the PAB model.

2.1.2 Sampling Positions

In this part, we will explain how we generate the start positions of the simulated reads based on the PAB model derived from the previous step. For every transcript, we create a transcript probability distribution, as defined in the following definition.

Definition 5 (Transcript probability distribution, TPD). For a given transcript, the transcript probability distribution describes the probability of producing a read at the current position in relation to the 3' end.

For each transcript, we will generate the start positions for simulated reads based on the TPD. For a given transcript, the corresponding TPD is defined in Equation (4). Let i represent the distance to the 3' end of the transcript, and m represent the number of PARs between the current location and the 3' end. The term d_{ij} is the distance from position i to the j-th PAR, and c is the weight factor that determines the influence of the poly(A)-tail at the 3' end of the transcript. We calculate the probabilities based on the PAB model, as shown in Equation (5).

$$TPD(x = i) = \begin{cases} cp_i + \frac{1-c}{m}\sum_{j=1}^{m} p_{d_{ij}}, if\ m \ge 1 \\ p_i, \qquad\qquad\qquad if\ m = 0 \end{cases} \quad (4)$$

$$p_i = PAB(x = i) \quad (5)$$

According to the definition of TPD, the probability of generating a read at the current location is affected by two components. The first one is the poly(A)-tail at the 3' end of the transcript, while the second one is the PARs inside the transcript. If the number of PARs $m = 0$, TPD is only affected by the first component, and the probability is directly calculated based on the PAB model. However, if the number of PARs $m \geqq 1$, the internal PARs of the transcript will also have a considerable influence on TPD. In this case, we should take both components into account.

In Equation (4), c serves as a weight factor, which is used to control the influence of the two components. Specifically, when an internal PAR is long, we would consider giving more weight to the bias caused by the PAR instead of the bias resulting from the poly(A)-tail. In this paper, we set $c = 0.5$ due to the fact that most transcripts in the training dataset do not have long internal PARs.

To summarize, we calculate TPD based on the PAB model, which takes both the poly(A)-tail and the internal PARs into account. Through sampling from the TPD for each transcript, we generate a list of start positions of simulated reads. Generally, an amplification factor is applied to ensure the read coverage for every transcript. To be more specific, this parameter is mathematically defined as the number of sampling iterations from the TPD for individual transcripts during stochastic read generation.

2.2 Fragmentation and Sequencing

In this part, the transcripts are fragmented, and sequencing reads are generated from the start positions we obtained in the previous step. As we have introduced the typical bias PAB for tag-based scRNA-seq methods in the model, we can adopt a bulk RNA-seq simulator to perform the following part of the simulation process. Polyester, a widely used bulk RNA-seq simulator, is used to produce the simulated reads.

Firstly, Polyester generates short fragments for every transcript based on the list of start positions we obtained from the corresponding TPD before. Fragment lengths are stochastically generated following a normal distribution $N(100,10^2)$ as Polyester's default configuration. Subsequently, the simulator performs directional synthesis by extracting single-end reads from the 5'-terminal R nucleotides of each fragment, where R corresponds to the user-defined sequencing read length.

In Drop-seq experiments, sequencing outputs adopt a dual-read structure: the initial read embeds both cellular identifiers (barcodes) and unique molecular identifiers (UMIs), whereas the subsequent read captures partial transcript sequences synthesized using Polyester. Given that Drop-seq protocols predominantly utilize Illumina platforms, we implemented Polyester's statistical error profile: an empirically derived model trained on Illumina sequencing data. This framework quantifies positional error rates by calculating substitution probabilities for each nucleotide base across read positions. Although our pipeline defaults to this platform-specific error simulation, Polyester's modular architecture permits the integration of alternative error models. Final outputs are formatted as FASTA-compliant single-read records, ensuring compatibility with standard downstream analysis tools.

2.3 Barcode Generation

Since the reads produced by Drop-seq have a dual-read structure, the simulated results should also consist of two parts: the cell barcode and molecular barcode (UMI), as defined in the following definitions.

Definition 6 (Cell Barcode). A cell barcode is a short, unique nucleotide sequence incorporated into each mRNA molecule during scRNA-seq to label transcripts originating from the same cell.

Definition 7 (Unique Molecular Identifier, namely UMI). A Unique Molecular Identifier (UMI) is a short, random sequence of nucleotides added to each mRNA molecule during the reverse transcription step in scRNA-seq.

UMIs are crucial for distinguishing original mRNA molecules from PCR duplicates. The cell barcodes are identical across all primers on one bead but differ across beads, allowing identification of the cell of origin. In contrast, each primer receives a unique UMI, which distinguishes between different mRNA molecules. In Drop-seq experiments, both the cell barcodes and UMIs are generated through "split-and-pool" synthesis cycles, [6].

The simulation framework implements a probabilistic nucleotide assignment strategy for biological indexing. Cellular identification relies on 12-nucleotide combinatorial sequences stochastically generated through permutations of canonical DNA bases (adenine, guanine, cytosine, thymine), whereas transcript-level tracking employs 8-mer unique molecular identifiers (UMIs) constructed via analogous randomization. Barcode allocation follows computational mapping: cellular signatures are uniquely assigned according to the input transcript abundance matrix, while UMIs are deterministically linked to individual RNA molecules. This design ensures systematic traceability, where sequencing reads exhibit UMI homogeneity within identical transcriptional units (intra-molecule conservation) and UMI heterogeneity across distinct molecular origins (inter-molecule divergence).

3 Datasets and Experimental Settings

3.1 Datasets

For model development and validation, we utilize publicly accessible Drop-seq datasets obtained from the Gene Expression Omnibus (GEO) repository. These curated datasets provide standardized benchmarks for assessing simulation fidelity across diverse cellular contexts.

The experiments were conducted on two datasets (GSM1544798 and GSM2177570). The first dataset contains samples of mouse and human data. We trained our model only using the mouse data in the first dataset, and the remaining human samples were used for testing. To be more specific, as introduced in [25], multiple isoform genes introduce ambiguities and uncertainties in expression analysis, so we restrict the training to single isoform transcripts. After filtering, we retain 401 transcripts for training our model. The second dataset contains mouse data, and it is also used for testing the effectiveness of the proposed method. In other words, we will demonstrate the performance of our model on two datasets in the following section.

3.2 Data Preprocessing

As for the Drop-seq data, data pre-processing follows the Drop-seq core computational protocol outlined by [6]. Reads are aligned using the RNS-seq aligner STAR, [26]. The reference genome and isoform annotations are based on hg19 for human cells and mm10 for mouse cells.

For the simulated data, data preprocessing is also required before we evaluate the performance of the model. Specifically, we must perform the same alignment process after obtaining the simulated reads. The reason for this is that the aligner could make mistakes during the alignment operation. For example, for transcripts with similar sequences, reads could be mapped into the wrong locations. Some reads might be discarded if the aligner cannot determine which transcript they are generated from. To account for these alignment errors, it is essential for us to align the simulated reads before evaluation. The alignment tool STAR is also applied in this step.

After alignment, discrepancies may arise between the total number of reads on a given transcript before alignment and after alignment. Therefore, we must normalize the data before evaluation. The normalized read number n_i at position i is determined according to the following formula.

$$n_i = \frac{\sum p}{\sum a} a_i \tag{6}$$

Here, $\sum p$ is the number of reads before alignment, and $\sum a$ is the number of reads after alignment. a_i is the number of reads at the corresponding position. This procedure should correct for the biases from the alignment process, and offer a more accurate means of evaluating our model.

3.3 Performance Evaluation

In this subsection, we will introduce how we evaluate the performance of the proposed simulator Ds-Sim, and the evaluation metrics we adopt. Before the evaluation process, we applied a sliding window strategy (window size of 100) as a preprocessing step to the read distribution. This procedure will tolerate some positional biases for evaluation. To be more specific, we group adjacent nucleotides into bins and calculate the average read count within each bin. This process is illustrated in Figure 2.

Following distribution generation, two quantitative metrics are employed to assess the statistical congruence between simulated outputs and experimental Drop-seq data profiles:

1. **Pearson Correlation Coefficient:** The Pearson correlation coefficient describes the similarity between the simulated data and the real Drop-seq data. The value ranges between (0,1). A higher Pearson value means that the two distributions are more similar. The calculation of Pearson is defined in Equation (7).

$$\text{Pearson} = \frac{N\sum xy - (\sum x * \sum y)}{\sqrt{N\sum x^2 - (\sum x)^2}\sqrt{N\sum y^2 - (\sum y)^2}} \tag{7}$$

2. **Bray-Curtis (BC) Distance:** The second metric is the Bray-Curtis (BC) distance, defined in Equation (8). The metric value is between (0,1). Opposite to Pearson, a lower BC distance means the two distributions are more similar.

$$\text{BC distance} = \frac{\sum(|x-y|)}{\sum(x+y)} \tag{8}$$

Both metrics provide a means of quantitatively evaluating the closeness of the simulated distributions to the real data, with higher scores indicating better performance.

4 Experiments

4.1 Comparison with the Baseline Model

In this subsection, we will demonstrate the performance of the proposed simulator Ds-Sim compared with a baseline model. Most current simulators adopt a uniform read distribution for each transcript [7], which performs well for bulk RNA-seq methods but not for tag-based scRNA-seq methods, for example, Drop-seq. In order to accurately model the bias introduced by the priming process, more reads should be generated from the positions closer to PARs. For most circumstances, the dominant PAR for a transcript is the poly(A)-tail at its 3' end, which results in a significant peak in the read distribution, [7]. Equation (9) serves as the baseline method to model this trend. Here, i represents the distance to the 3' end of the transcript, and $P(x)$ represents the sampling probability. It is obvious that the baseline model will generate more reads as the start position is closer to the 3' end of a transcript.

$$P(x = i) = \frac{1}{i} \tag{9}$$

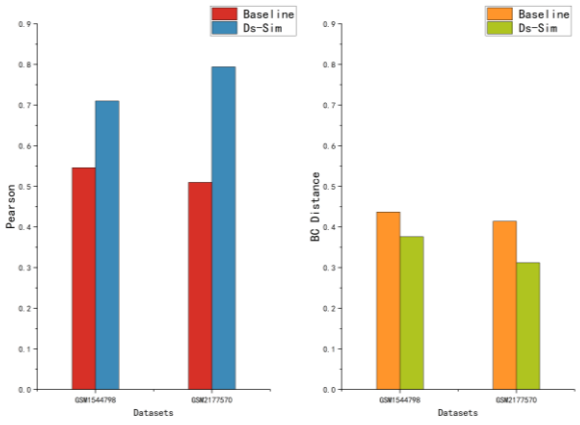

Fig. 3: Comparison of the baseline and the proposed Ds-Sim in terms of Pearson and BC distance

Figure 3 demonstrates the experiment results of the baseline model and the proposed Ds-Sim in terms of Pearson and BC distance on the experimental datasets. It is obvious that the proposed Ds-Sim achieves a higher Pearson value and a lower BC distance. On the first dataset, the Pearson value of Ds-Sim is nearly 30% higher than that of the baseline model; on the second dataset, it is 56% higher. As for the BC distance, the BC distance of Ds-Sim is 17% lower than the distance of the baseline model on the first dataset, and 25% lower on the second dataset. The proposed Ds-Sim shows a great improvement in terms of both metrics compared to the baseline model.

In Figure 4 (Appendix), we demonstrate the read distribution of three example transcripts selected from the experiment datasets. We plot the read distributions of the baseline model, the proposed Ds-Sim and the real Drop-seq data in each graph. As we can observe from the graphs, all transcripts demonstrate an obvious PAB, with a significant peak on the left side, which is caused by the poly(A)-tail at the 3' end of the transcript. In most cases, as shown in the first two transcripts, the distribution obtained by Ds-Sim demonstrates a more similar shape to the real data, compared to the baseline model. The baseline model captures the overall trend of the distribution, but fails to capture the bias introduced by other potential factors in a Drop-seq experiment. However, there are exceptions where the baseline model obtains a better result, such as the third example transcript. There are several possible factors that may lead to this result, since the real Drop-seq experiment is very complex. For example, in some cases, an extremely large number of reads appear near the 3' end of the transcript, and the read distribution will be closer to the baseline model. Yet, this is quite a rare circumstance. As we can tell from the experiment results, the read distribution generated by Ds-Sim is more similar to the real Drop-seq data, and the proposed Ds-Sim achieves a better overall performance than the baseline model.

4.2 Comparison of Single-isoform Transcripts and Multi-isoform Transcripts

In this subsection, we will demonstrate the performance of Ds-Sim on single-isoform transcripts and multi-isoform transcripts. For genes with multiple isoforms, it is difficult to determine which transcript the corresponding read was originally sequenced from. Most gene quantification tools are especially burdened by the multiple isoform issue and perform poorly when handling problems in this field, [27]. Fortunately, the proposed Ds-Sim achieves a satisfying performance on both single-isoform transcripts and multi-isoform transcripts.

Figure 5 demonstrates the experiment results of single-isoform transcripts and all transcripts in terms of Pearson and BC distance. From observation, the proposed Ds-Sim achieves a slightly better performance on single-isoform transcripts, but still obtains a satisfying result on the whole dataset.

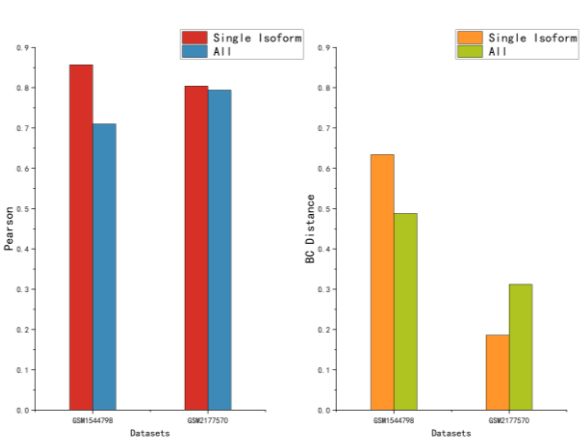

Fig. 5: Comparison of single-isoform transcripts and all transcripts in terms of Pearson and BC distance

On the first dataset, the Pearson value of single-isoform transcripts is 17% higher than that of all transcripts; on the second dataset, it is only 1.2% higher. As for the BC distance, the BC distance of single-isoform transcripts is 23% higher on the first dataset, and 40% lower on the second dataset.

All experiment results are acceptable, with the highest Pearson of 0.86 and the lowest BC distance of 0.18. It is obvious that transcripts with multiple isoforms did not have a detrimental effect on the experiment results. On the second dataset, multi-isoform transcripts only cause a difference of 1.2% in Pearson value. Even on the first dataset, adding multi-isoform transcripts results in an improvement in BC distance. We should notice that, as we removed multi-isoform transcripts to avoid ambiguities in the data preprocessing process, the model is only trained using single-isoform transcripts from the first dataset. Therefore, it shows great learning ability, demonstrating satisfying flexibility and adaptability to multi-isoform cases. Overall, the proposed Ds-Sim achieves an effective performance on both single-isoform transcripts and multi-isoform transcripts.

4.3 Comparison of Different Species

In this subsection, we will evaluate the performance of the proposed simulator Ds-Sim on different species. We conducted simulation experiments for both mouse and human data. The mouse cell samples are from the dataset GSM2177570, and the human cell samples are from the dataset GSM1544798.

Figure 6 presents the experiment results on the two datasets in terms of Pearson and BC distance. From observation, we notice that the experiment results of the two species are quite similar, only with a slight difference. To be more specific, the Pearson value of the human data is about 4.8% higher than that of the mouse data.

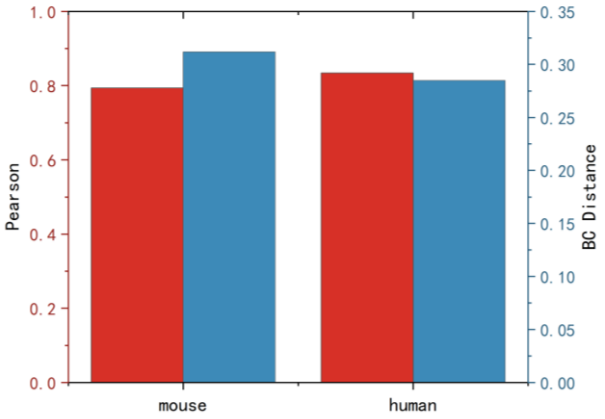

Fig. 6: Comparison of different species in terms of Pearson and BC distance

In terms of the BC distance, the BC distance of the human data is about 8.6% lower than that of the mouse data. It seems the experiment results on the human data are slightly better. There are several possible reasons for this. We infer this may be because the human samples are from GSM1544798, so they share similar characteristics to the training data. For example, they were obtained under the same experiment condition, and they have the same read length, etc. Still, the proposed Ds-Sim achieves satisfying simulation results on both species, with a Pearson value around 0.8 on both datasets, which suggests that the distribution of simulated reads obtains similar shape and features to the real Drop-seq reads. Furthermore, human samples are completely new to the simulator, since it is only trained on mouse samples. Despite that, the proposed Ds-Sim demonstrates great adaptability to an entirely new species. Overall, the proposed Ds-Sim is able to model the PAB caused by the sequencing process, and produce valid simulation data for different species.

5 Conclusion

In this paper, we present a novel RNA read simulator, namely Ds-Sim, which can simulate reads of Drop-seq data that accurately capture the poly(A)-bias inherent in single-cell RNA sequencing (scRNA-seq) data. Unlike conventional RNA-seq simulation methods that assume uniform read distributions, our method explicitly models the skewed distribution caused by poly(A) priming, resulting in a more biologically realistic read distribution. Furthermore, by introducing a mixture

model that dynamically adjusts the read sampling probability based on read length and distance to poly(A)-regions, we capture nuanced positional biases that better reflect real Drop-seq experiments. In addition, the integration of realistic barcode and UMI generation, combined with empirical sequencing error models, ensures that the simulated data closely mimics real Drop-seq data at multiple levels. Our approach not only improves the quality of simulated reads but also provides a robust framework for benchmarking and evaluating new computational methods designed for Drop-seq data analysis.

Our model can be further enhanced by incorporating additional sources of bias, such as GC content or transcript secondary structures, which may also affect read distributions. Additionally, extending the model to accommodate other single-cell sequencing technologies, such as 10x Genomics or Smart-seq, would broaden its applicability. Future work could also explore improving the barcode synthesis process to capture more complex cell barcode structures and error profiles.

Declaration of Generative AI and AI-assisted Technologies in the Writing Process

The authora wrote, reviewed and edited the content as needed and they have not utilised artificial intelligence (AI) tools. The authors take full responsibility for the content of the publication.

References:

[1] Wu, A.R., Neff, N.F., Kalisky, T., Dalerba, P., Treutlein, B., Rothenberg, M.E., Mburu, F.M., Mantalas, G.L., Sim, S., Clarke, M.F. and Quake, S.R., 2014. Quantitative assessment of single-cell RNA-sequencing methods. *Nature methods,* 11(1), pp.41-46.

[2] Patel, A.P., Tirosh, I., Trombetta, J.J., Shalek, A.K., Gillespie, S.M., Wakimoto, H., Cahill, D.P., Nahed, B.V., Curry, W.T., Martuza, R.L. and Louis, D.N., 2014. Single-cell RNA-seq highlights intratumoral heterogeneity in primary glioblastoma. *Science,* 344(6190), pp.1396-1401.

[3] Zheng, C., Zheng, L., Yoo, J.K., Guo, H., Zhang, Y., Guo, X., Kang, B., Hu, R., Huang, J.Y., Zhang, Q. and Liu, Z., 2017. Landscape of infiltrating T cells in liver cancer revealed by single-cell sequencing. *Cell,* 169(7), pp.1342-1356.

[4] Paul, F., Arkin, Y.A., Giladi, A., Jaitin, D.A., Kenigsberg, E., Keren-Shaul, H., Winter, D., Lara-Astiaso, D., Gury, M., Weiner, A. and David, E., 2015. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell,* 163(7), pp.1663-1677.

[5] Ramsköld, D., Luo, S., Wang, Y.C., Li, R., Deng, Q., Faridani, O.R., Daniels, G.A., Khrebtukova, I., Loring, J.F., Laurent, L.C. and Schroth, G.P., 2012. Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nature biotechnology,* 30(8), pp.777-782.

[6] Macosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A.R., Kamitaki, N., Martersteck, E.M. and Trombetta, J.J., 2015. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell,* 161(5), pp.1202-1214.

[7] Young, M. D., & Behjati, S. (2018). SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data. *Genome Biology*, 19(1), 210.

[8] Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., Lönnerberg, P. and Linnarsson, S., 2014. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature methods,* 11(2), pp.163-166.

[9] de Klerk, E., Den Dunnen, J.T. and 't Hoen, P.A., 2014. RNA sequencing: from tag-based profiling to resolving complete transcript structure. *Cellular and molecular life sciences,* 71, pp.3537-3551.

[10] Ziegenhain, C., Vieth, B., Parekh, S., Reinius, B., Guillaumet-Adkins, A., Smets, M., Leonhardt, H., Heyn, H., Hellmann, I. and Enard, W., 2017. Comparative analysis of single-cell RNA sequencing methods. *Molecular cell,* 65(4), pp.631-643.

[11] Aggarwal, A. K. (2023). A review on genomics data analysis using machine learning. *WSEAS Transactions on Biology and Biomedicine*, vol.20, pp.119-131, https://doi.org/10.37394/23208.2023.20.12.

[12] Torgano, F., Cavalleri, E., Gliozzo, J. E. S. S. I. C. A., Stacchietti, F. E. D. E. R. I. C. O., Saitto, E. M. A. N. U. E. L. E., Mesiti, M. A. R. C. O., & Valentini, G. (2024). RNA Knowledge Graph Analysis via Embedding Methods. *WSEAS Transactions on Biology and Biomedicine*, vol.21, pp.302-312, https://doi.org/10.37394/23208.2024.21.30.

[13] Kisang Kwon, Eun-Ryeong Lee, Kyung-Hee Kang, Seung-Whan Kim, Hyewon Park, Jung-Hae Kim, An-Kyo Lee, O-Yu Kwon (2023). Identification and Expression Analysis of a

cDNA Encoding Cyclophilin A from Gryllus bimaculatus (Orthoptera: Gryllidae). *WSEAS Transactions on Environment and Development*, vol.19, pp.457-464, https://doi.org/10.37394/232015.2023.19.43.

[14] Huang, W., Li, L., Myers, J.R. and Marth, G.T., 2012. ART: a next-generation sequencing read simulator. *Bioinformatics,* 28(4), pp.593-594.

[15] Balzer, S., Malde, K., Lanzén, A., Sharma, A. and Jonassen, I., 2010. Characteristics of 454 pyrosequencing data—enabling realistic simulation with flowsim. *Bioinformatics,* 26(18), pp.i420-i425.

[16] Angly, F.E., Willner, D., Rohwer, F., Hugenholtz, P. and Tyson, G.W., 2012. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic acids research,* 40(12), pp.e94-e94.

[17] Shcherbina, A., 2014. FASTQSim: platform-independent data characterization and in silico read generation for NGS datasets. *BMC research notes,* 7, pp.1-12.

[18] Frazee, A.C., Jaffe, A.E., Langmead, B. and Leek, J.T., 2015. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics,* 31(17), pp.2778-2784.

[19] Escalona, M., Rocha, S. and Posada, D., 2016. A comparison of tools for the simulation of genomic next-generation sequencing data. Nature Reviews Genetics, 17(8), pp.459-469.

[20] Zappia, L., Phipson, B. and Oshlack, A., 2017. Splatter: simulation of single-cell RNA sequencing data. *Genome biology,* 18(1), p.174.

[21] L. Lun, A.T., Bach, K. and Marioni, J.C., 2016. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome biology,* 17, pp.1-14.

[22] Lun, A.T. and Marioni, J.C., 2017. Overcoming confounding plate effects in differential expression analyses of single-cell RNA-seq data. *Biostatistics,* 18(3), pp.451-464.

[23] Vallejos, C.A., Marioni, J.C. and Richardson, S., 2015. BASiCS: Bayesian analysis of single-cell sequencing data. *PLoS computational biology,* 11(6), p.e1004333.

[24] Rizzetto, S., Eltahla, A.A., Lin, P., Bull, R., Lloyd, A.R., Ho, J.W., Venturi, V. and Luciani, F., 2017. Impact of sequencing depth and read length on single cell RNA sequencing data of T cells. *Scientific reports,* 7(1), p.12781.

[25] Li, B., Ruotti, V., Stewart, R.M., Thomson, J.A. and Dewey, C.N., 2010. RNA-Seq gene expression estimation with read map** uncertainty. *Bioinformatics,* 26(4), pp.493-500.

[26] Dobin, A., Davis, C.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M. and Gingeras, T.R., 2013. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics,* 29(1), pp.15-21.

[27] Arzalluz-Luque, Á. and Conesa, A., 2018. Single-cell RNAseq for the study of isoforms—how is that possible?. *Genome biology,* 19, pp.1-19.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

- Wenshan Li led the entire research, including algorithms, experiments, and writing.
- Wenbo Fang, Ao Liu, Beibei Li, Junjiang He, and Hongxia Wang analyzed the experimental data.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

This work is partially supported by the National Key Research and Development Program of China (2020YFB1805400), and the National Natural Science Foundation of China (2032002, 62272331, 62372313, 62402330).

Conflict of Interest

The authors have no conflicts of interest to declare.

APPENDIX

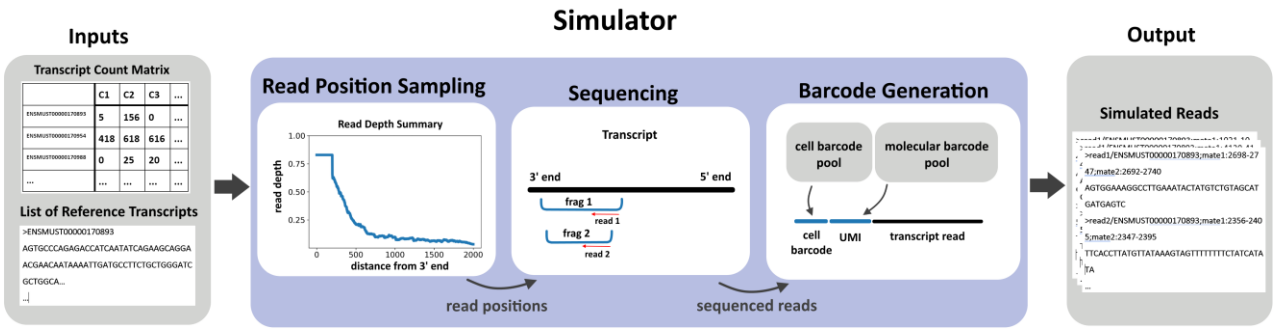

Fig. 1: The method framework

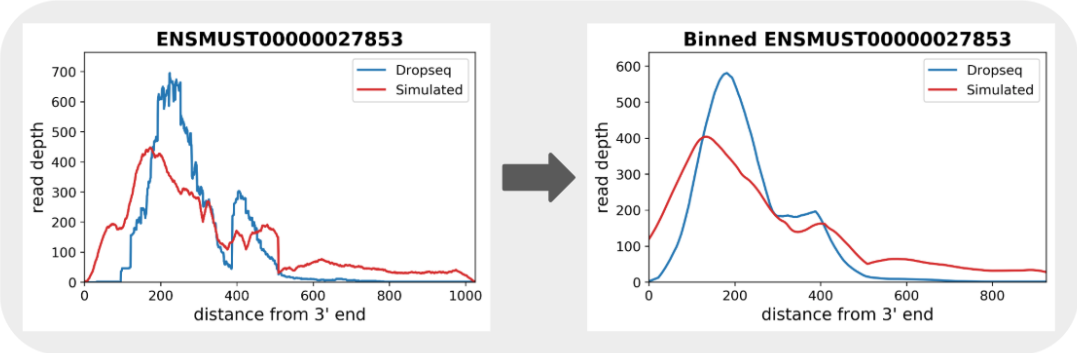

Fig. 2: The original read distribution (left) and the distribution processed by the sliding window (right)

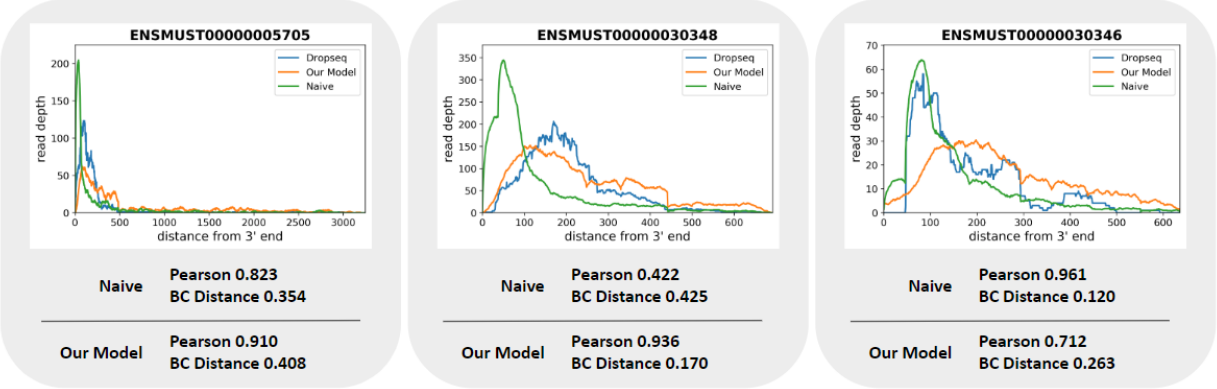

Fig. 4: The read distributions of real Drop-seq data, the baseline model and the simulated data generated by the proposed Ds-Sim