scReadSim: a single-cell multi-omics read simulator

Guanao Yan [1] and Jingyi Jessica Li [1,2,3,4,*]

Abstract

Single-cell sequencing technologies emerged and diversified rapidly in the past few years, along with the successful development of many computational tools. Realistic simulators can help researchers benchmark computational tools. However, few simulators can generate single-cell multi-omics data, and none can generate reads directly. To fill in this gap, we propose scReadSim, a simulator for single-cell multi-omics reads. Trained on real data, scReadSim generates synthetic sequencing reads in BAM or FASTQ formats. We deployed scReadSim on a sci-ATAC-seq dataset and a single-cell multimodal dataset to show the resemblance between synthetic and real data at the read and count levels. Moreover, we show that scReadSim allows user-specified ground truths of accessible chromatin regions for single-cell chromatin accessibility data generation. In addition, scReadSim is flexible for allowing varying throughputs and library sizes as input parameters to guide experimental design.

[1] Department of Statistics, University of California, Los Angeles, CA 90095-1554
[2] Department of Human Genetics, University of California, Los Angeles, CA 90095-7088
[3] Department of Computational Medicine, University of California, Los Angeles, CA 90095-1766
[4] Department of Biostatistics, University of California, Los Angeles, CA 90095-1772
[*] To whom correspondence should be addressed. Email: jli@stat.ucla.edu

Introduction

The rapid development of single-cell sequencing technologies makes it possible to depict the single cells over genomic, epigenomic, and transcriptomic multi-layers[1]. The single-cell genomic sequencing technologies, so called single-cell DNA sequencing (scDNA-seq) methods, helps elucidate the genetic heterogeneity under cellular resolution. The single-cell epigenomic sequencing technologies measures the distinct epigenetic characteristics among individual cells: DNA modifications (scRRBS[2], scBS-seq[3]), histone modifications (scChIP-seq[4, 5], CUT&Tag[6, 7]), chromatin accessibility (scATAC-seq[8], sciATAC-seq[9]), and three-dimensional genome architecture (single-cell Hi-C[10]). Specifically, single-cell Assay for Transposase-Accessible Chromatin using sequencing (scATAC-seq)[8] measures the chromatin accessibility over thousands of cells. This sequencing technology has been widely used to investigate the behavior of regulatory elements, such as nucleosome positions and transcription factor binding sites. The single-cell transcriptomic sequencing technology, also known as scRNA-seq, reveals the genome expression levels within individual cells, helping resolve cellular heterogeneity[11] and enabling the study of monoallelic gene expression[12], splicing patterns[13], and noise during transcriptional responses[14]. Given the prosperity of sequencing technologies, more computational algorithms and analysis tools emerged in the past decades[15]. Therefore, making choices from tremendous computational tools remains the most essential but difficult question for researchers to answer.

Simulators generating synthetic data keep the ground truth for users, which could help researchers benchmark various computational tools like evaluating the performance or quantifying the methods' scalability in terms of computational efficiency. There have been various read simulators built for single-cell genomic sequencing data. SimSCScTree[16], SCSIM[17], SCSsim[18] and CellCoal[19] are able to generate sequencing reads under single-cell resolution to benchmark analysis tools for single nucleotide variants(SNVs)[20] or copy number aberrations (CNAs)[21]. In contrast, few methods can simulate sequencing reads for single-cell epigenomic and transcriptomic data. Categorized by the specific sequencing technologies, scDesign[22], scDesign2[23], splatter[24], ZINB-WaVE [25] and many other methods have been developed for scRNA-seq; while only two methods simATAC[26] and SCAN-ATAC-Sim[27] have been created for scATAC-seq.

Specifically, [28] is the only read simulator built for scRNA-seq, which starts from a user-specified count matrix and a list of reference transcripts corresponding to the count matrix. This method models the distances between the reads' starting positions to the poly-A sites of gene transcripts from a training dataset. Given the pre-trained model and the input count matrix, it samples reads' positions according to this parametric model for each gene transcript. However, this method lacks authenticity. It uses one single model to measure read positions for all gene transcripts homogeneously and only generates reads for restricted areas like the gene transcripts but simulates no read for the complementary regions of the reference genome. More importantly, it fails to provide software for public usage. The necessity of read simulators also applies to scATAC-seq.

SCAN-ATAC-Sim [27] is the only simulator that generates scATAC-seq reads but only takes bulk samples as input. By separating the reference genome into multiple regions, SCAN-ATAC-Sim generates synthetic reads for each cell by first sampling a certain number of regions with probability proportional to the read coverage of bulk samples. Then it randomly samples one single read for each sampled region from the bulk data. The amount of regions (reads) sampled for each cell is decided based on two user-input parameters: library size per cell N_c and signal ratios ρ, which alters the read proportions between peak regions and non-peak regions. SCAN-ATAC-Sim can preserve the cell-type level information from bulk samples by sampling reads separately for different cell types. However, there are three aspects that SCAN-ATAC-Sim differs from our simulator, scReadSim, proposed below:

- SCAN-ATAC-Sim is not designed for simulating realistic single-cell ATAC-seq data. Instead, it generates single-cell reads depending on the user-input parameters and an empirical distribution obtained from the bulk samples. For example, the read library size for each cell has a default value ranging from 1000 to $20{,}000$, and SCAN-ATAC-Sim also uses a default signal ratio 0.7 to alter the reads proportion between peaks and non-peak regions. However, our method incorporates scDesign2[23] to model the read counts through a joint parametric distribution among peaks and non-peaks and then generate the synthetic count matrix for sampling reads in the next steps. Hence, scReadSim can generate realistic single-cell synthetic reads in both read and count levels. In addition, scReadSim can preserve the correlation among features through a Gaussian copula structure within scDesign2, shown in our Results section. Furthermore, the parametric modeling of the counts' distribution provides users with a more flexible interface by allowing them to adjust, for example, the mean parameters of features.

- SCAN-ATAC-Sim generates one single read for each feature from one parental trial within one cell. However, this process might be unauthentic because it is still possible that multiple non-overlapped reads occur within a longer feature. In other words, the count matrix of scATAC-seq should have entries larger than two. In contrast, scReadSim samples reads according to the synthetic count matrix simulated from a parametric generative model estimated from the real data.

- SCAN-ATAC-Sim is a simulator solely designed for ATAC-seq data. However, scReadSim is a single-cell multi-omics read simulator targeting scRNA-seq and scATAC-seq.

Most single-cell genomic and epigenomic sequencing data simulators are based on the read count matrices. The count-level simulators take the real count matrix as the synthetic count matrix's input and output. Therefore, the low-level bioinformatics tools that start from FASTQ or BAM formats can not directly use the synthetic count matrices generated by these count-based simulators. For example, the peak calling methods (MACS3[29], HOMER[30], HMMRATAC[31], SEACR[32], PeakRanger[33]) use the aligned BAM files as input to find the chromatin accessibility

regions along the reference genome; the read alignment tools like bowtie2[34] and plenty of analysis protocols like SnapATAC[35] start from the unaligned FASTQ files. Hence, to provide ground truth for the low-level bioinformatics tools, developing a read simulator for single-cell epigenomic and transcriptomic sequencing data is necessary.

Single-cell sequencing data helps investigate cellular biological information through genomic, epigenomic, and transcriptomic measurements. This multi-layer interface stimulates the development of analysis tools that attempt to integrate information obtained independently from different layers[36, 37]. Seurat3[38] integrates scRNA-seq and scATAC-seq to reflect the cell type-specific regulatory loci. MOFA[39] reveals matched transcriptional and epigenetic changes within the cell differentiation by parallelly analyzing the single-cell RNA expression and DNA methylation data. An optimal solution to benchmark these single-cell multi-modalities methods would be using a simulator that can simultaneously generate multi-modal synthetic data. However, current simulators for single-cell sequencing data only generate a single modality. [40] claims to simulate single-cell multi-modal sequencing data. However, it works solely for scRNA-seq and does not generate synthetic data for multiple modalities. Therefore, the restricted number of modalities generated by existing simulators calls for a single-cell multi-omics sequencing data simulator.

Hence, we propose scReadSim, the first sequencing reads simulator for single-cell multi-omics data. It takes the BAM file of single-cell sequencing data as input and generates the synthetic reads in the BAM or FASTQ formats (Figs. 1-2). We demonstrate that scReadSim generates synthetic data resembling the real data in both counts and read levels. We further justify the authenticity of biological information in scReadSim synthetic data by examining the chromatin open regions identified using peak calling methods. In addition, we include two applications of scReadSim. First, scReadSim allows users to specify arbitrary chromatin open regions for the simulated scATAC-seq reads. Second, users could specify any amount of synthetic cells and sequencing depth for their single-cell gene expression data when deploying scReadSim. We also include a benchmark study using the synthetic scATAC-seq data generated by scReadSim to investigate four popular peak callers: MACS3[29], HOMER[30], HMMRATAC[31] and SEACR[32].

Results

An overview of scReadSim

This section shows that scReadSim can simulate realistic single-cell sequencing data for multiple omics, including scRNA-seq and scATAC-seq. We deploy scReadSim onto the single-cell combinatorial indexing assay for transposase accessible chromatin with sequencing (sci-ATAC-seq) data[41] measured on mouse bone marrow tissue, the gene expression data and chromatin accessibility data of 10x single-cell multimodal dataset, see Figs. 3-5 and Supplementary Figs. S1–S6. We compare the synthetic data generated by scReadSim with the real data under both read and count levels. In addition, we show that users could utilize scReadSim to specify any chromatin

open regions for the synthetic scATAC-seq data, enabling users to freely design the ground truth for benchmarking computational methods (Fig. 6 and Supplementary Figs. S7-S8). In addition, scReadSim is flexible for allowing varying throughputs and library sizes as input parameters to guide experimental design, see Fig. S9.

Synthetic single-cell single omic data generated by scReadSim resembles real data in read level

The synthetic reads generated by scReadSim are output in FASTQ files or a BAM file by aligning the FASTQ files onto the reference genome. This section includes three comparison parts to illustrate that the scReadSim synthetic data mimics real data under the read level. Firstly, we compare the real and synthetic BAM files through quantitative measurements, including k-mer spectrum, fragment length distribution, error rate per base within reads, and the Reads Per Kilobase Million (RPKM). Secondly, we directly compare the read coverage of synthetic and real BAM files using the IGV genome browser[42]. This visualization depicts how the pooled single-cell reads align with the reference genome and illustrates the bulk-level similarity of read coverage between the two BAM files. Thirdly, we further verify the authenticity of the synthetic scATAC-seq data by deploying the peak caller MACS3 onto both real and synthetic BAM files and comparing the peak calling results. The results from above comparisons are summarized for 10x scATAC-seq (Figs. 3–4), sci-ATAC-seq (Figs. S1–S2) and 10x scRNA-seq (Fig. S4).

Sequencing reads simulators like ReSeq[43] for Illumina sequencing technologies usually focus on mimicking the biological traits of real reads, including the coverage, quality, and base calling. Similarly, we compare scReadSim reads with real reads through the following quantitative measurements from BAM files: k-mer spectrum, fragment length distribution, error rate per base within reads, and the Reads Per Kilobase Million (RPKM). These summary statistics directly reflect the reads' sequencing information. First, we examine the k-mer spectrum of real and synthetic reads, which measures the occurrences of substrings of length k within the read sequences. In Figs. 3A, S1A, S4A, the x-axis refers to the amount of frequency occurred for a specific k-mer, while y-axis represents the number of unique k-mers that have this frequency. We compute 11 to 41-mer for both synthetic and real reads and find that their k-mer spectrum pattern resembles each other for multi-omics, including scATAC-seq and scRNA-seq. Second, for paired-end sequencing data, the distance between starting positions of two mates, the so-called fragment length, also reflects information about the chromatin states. For example, the shorter fragment is, the higher probability that the chromatin region covered by the fragment is open, i.e., not wrapping any nucleosome. Figs. 3B, S1B, S4B compare the fragment length distribution between the real and synthetic BAM files, which also verifies that scReadSim is able to produce realistic reads. Third, as one symbolic trait of the realistic sequencing reads, the sequencing errors occur in the standard Illumina sequencing of genomic DNA, among which the substitution errors are most common. In Figs. 3C, S1C, S4C, we show that scReadSim can learn and reproduce substitution errors from the

real data. Fourth, to summarize the read coverage over the reference genome quantitatively, we use the peak calling method MACS3 to call peaks on both real and synthetic BAM files. scReadSim separates the reference genome into foreground features, like gene transcript regions for scRNA-seq or peaks for scATAC-seq, and background features covering the reference genome's remaining areas. We take the peaks identified by MACS3 as the foreground features and the complementary regions as background features. Thus we have foreground (background) features for synthetic and real BAM files. Then we compute the RPKM (See Method) for peak regions (foreground features) and the complementary regions (background features) separately. We find that the RPKM of the scReadSim BAM file resembles the real data regarding both foreground and background features, which justifies that the synthetic BAM file preserves similar biological signals from the real BAM file (Figs. 3D, S1D). We also observe that the RPKM values of the foreground features are significantly larger than those of the background features, which further validates our categorization of the foreground and background features along the reference genome.

The direct comparison regarding the read coverage for the synthetic and real BAM files using the IGV genome browser are summarized in Figs. 4C, S2C, S4C. We find that the single-cell sequencing reads simulated by scReadSim mimic the real data at a bulk level for foreground and background features.

For scATAC-seq, we further verify that the synthetic BAM file generated by scReadSim preserves the chromatin open information observed in the real peak regions. By comparing the peak calling results, we find that most real peaks can be identified by deploying MACS3 on the scReadSim BAM file (Figs. 4A, S2A). In addition, we also plot the ROC (receiver operating characteristic) curve and precision-recall curve by taking the real peaks as truth. Specifically, we vary a threshold value over the p-values of peaks identified by MACS3 deployed on the synthetic BAM file. The curves imply that MACS3 does an excellent job in identifying true peak regions using the synthetic BAM file, while the precision rate tends to be low when the recall is large (Figs. 4B, S2B).

Synthetic single-cell single omic data generated by scReadSim resembles real data in count level

A count matrix is a numerical matrix with entries representing the number of reads overlapping each feature for every individual cell. The rows correspond to features, and the columns represent cells. Typically, the features could be chromatin open regions (peaks) for scATAC-seq or gene transcript regions for scRNA-seq. Given the count matrix processed from the input BAM file, scReadSim first uses scDesign2 to fit a probabilistic generative model from the real count matrix and then simulates a synthetic count matrix. The synthetic count matrix would guide sampling reads from the real BAM file for each synthetic cell in output data. Hence a realistic synthetic count matrix is essential in simulating single-cell sequencing reads which preserve the real biological signals.

We compare foreground features' synthetic and real count matrices throughout the following aspects. Firstly, we utilize the following summary statistics to describe the distribution of the count matrix entries: for every feature, we calculate the mean, variance, coefficient of variance, and zero proportion across cells; for every cell, we calculate the library size and zero proportion across features. Secondly, we sum the read counts over all cells for each feature and examine the relationship between the marginal count values and the locations of features. Specifically, we calculate the correlation of marginal count values between scReadSim synthetic and real data among 100 windows along the reference genome. Each window contains the same number of adjacent foreground features except for the last window. The x-axis refers to the window index, from 1 to 100 covers all foreground features along the reference genome. The negative control group in red color represents the correlation of real marginal counts between random pairs of windows. Specifically, we sample windows with replacement from another 99 windows to form 100 window pairs by fixing one window. Then we compute the correlation of real marginal counts between each window pair and report the mean and standard deviation of the 100 correlation values for the fixed window index. Thirdly, to illustrate that scReadSim preserves the correlation among the features, we select 100 top-expressed features from the real count matrix and use heatmaps to show the correlation among these features for both real and synthetic count matrices. Fourthly, we utilize principal component analysis (PCA), and uniform manifold approximation and projection (UMAP)[44, 45] to find the low-dimensional embedding of the combined real and synthetic count matrices. We label the cells with either the dataset sources (synthetic cells generated by scReadSim or real cells) or the cell types pre-defined in the original dataset (or cell clusters obtained through the scReadSim synthetic count matrix generation procedure). When labeling cells with the dataset sources, we also report the median of the integration local inverse Simpson's index (miLISI) proposed by [46] to quantify the mixing levels of synthetic and real cells (see Method). The miLISI value (ranging from 1 to 2) will be close to 2 if a perfect mixture exists between the synthetic and real cells.

By deploying scReadSim onto multi-omic sequencing technologies, we find that scReadSim could generate synthetic data resembling the real data in the count level.

First, throughout the six statistics that we summarize from the count matrices, we find that the synthetic count matrix mimics the real count matrix in both feature and cell levels, see Fig. 5A and Supplementary Figs. S3A, S5A. The feature level, including mean, variance, coefficient of variance, and zero proportion, describes each feature's marginal distribution of count entries. We find that the synthetic count matrix could maintain the same distribution traits and preserve a similar feature sparsity for both scRNA-seq and scATAC-seq. The cell level statistics, including zero proportion and sequencing depth for each cell, also indicate that scReadSim mimics the real data.

Second, we examine the correlation between the synthetic and real marginal counts over foreground features along the genome by splitting the features into 100 windows. From Figs. 5B, S3B, S5B we find that the correlation between the synthetic and real count values are always close to 1, in contrast to the negative control group, which concentrates around 0. This observation

indicates that scReadSim generates realistic count values for the foreground features along the reference genome.

Third, we use a heatmap to show the correlation among the top expressed features obtained from the real count matrix and compare it with the synthetic data produced by scReadSim. The results (Figs. 5C, S3C, S5C) imply that scReadSim could preserve the correlation structure among features from the real count matrix.

Fourth, by examining the 2D embeddings of the combined real and synthetic count matrices, we observe a perfect mixture between synthetic and real cells (Figs. 5D, S3D, S5D Left). In other words, the synthetic cells generated by scReadSim mimic the real cells in the low-dimensional embedding. This observation is also verified by the miLISI values close to 2. In addition, by labeling the cells with cell types (clusters), we find a clear separation among different cell types, which also verifies that the synthetic data generated by scReadSim preserves the original biological variation to distinguish different cell clusters, see Figs. 5D, S3D, S5D Right.

Application 1: scReadSim generates synthetic scATAC-seq data based on user-specified chromatin open regions

The scATAC-seq data measures the chromatin accessibility along the genome. The chromatin open regions reflect the cell level sites related to the potentially undergoing gene transcription process. The most popular way to identify the chromatin open regions for scATAC-seq is to first pool single cells together as the so-called pseudo-bulk data and then use that as input of the bulk-based peak calling methods. The identified open regions are known as peaks in single-cell epigenetic data analysis. This section shows that in addition to generating synthetic reads that mimic the real data, scReadSim could also allow users to specify peak regions for their synthetic scATAC-seq reads. We show that the designed synthetic scATAC-seq data preserves the biological information from the real data for both the peaks (foreground) and background regions. By holding the ground truth of chromatin accessible locations, one could use the simulated BAM file to benchmark low-level bioinformatics tools like peak calling methods. We also use a user-input-peak synthetic scATAC-seq dataset generated by scReadSim to benchmark four popular peak calling methods: MACS3, HOMER, SEACR, and HMMRATAC.

To illustrate that scReadSim can let users freely define the chromatin open regions along the genome for their synthetic scATAC-seq reads, we input scReadSim with a sci-ATAC-seq BAM file and a manually created pseudo-peak regions set. Specifically, we truncate each gene transcript region from its transcription starting site (TSS) with a random length uniformly sampled from $[250, 550]$ base pairs. Then we use scReadSim to generate a synthetic scATAC-seq dataset based on the specified pseudo-peak regions. We use the IGV genome browser to directly visualize the read coverage of synthetic and real BAM files (Fig. S7). In addition, to validate the authenticity of the designed synthetic BAM file output by scReadSim, we utilize the following quantitative measurements of reads: the k-mer spectrum, fragment lengths distribution, substitution error rates

per base call, and RPKM for foreground and background features (Fig. S8).

The genome browser (Fig. S7) shows that the synthetic BAM file with a user-input peak set has a similar pattern in terms of the read coverage when compared with the real BAM file, especially when looking into the foreground and background features along the reference genome separately. The read coverage of the user-input pseudo-peaks significantly differs from the background features, resembling the distinctions between the read coverage of real chromatin open regions and complementary areas. The RPKM comparison further exemplifies this observation (Fig. S7D): the RPKM values for synthetic BAM over user-input peaks mimic the real chromatin open areas. They are excessively larger than their corresponding background features. The distinction of RPKM between foreground and background features also justifies the fact that the peak region should have a more extensive read coverage than its corresponding background features. Besides, the comparison of k-mer spectrum (Fig. S1A), fragment length distribution (Fig. S8B) and error rates per base call (Fig. S8C) also illustrate that scReadSim is able to preserve the sequence information from real data for the user-designed scATAC-seq synthetic data. In other words, the authenticity of synthetic reads based on the user-specified chromatin open regions guarantees that scReadSim can provide ground truths for evaluating the performance of computational methods.

We further include a benchmark study to show that our synthetic BAM file with ground truths (i.e., the user-input chromatin open regions) could help benchmark low-level bioinformatics tools. The task is to evaluate four of the most popular peak calling methods that take a BAM file as input. We include the third version of MACS3, which was initially designed for identifying transcript factor binding sites for ChIP-sequencing (ChIP-seq) data; SEACR, a highly selective peak caller designed for CUT&RUN datasets; HOMER, a peak caller for ChIP-seq but also widely used for ATAC-seq; HMMRATAC, a peak caller designed specifically for ATAC-seq data. We deploy these four peak callers with the default settings onto the synthetic BAM file generated by scReadSim, for which we truncate each gene transcript region as our designed chromatin open regions. Then we compare the peak calling results of the four peak callers with the ground truth, i.e., the input chromatin open regions, throughout the following four aspects. First, we calculate the RPKM values of the synthetic BAM file using five different feature spaces. The first feature space, User Input, includes foreground features representing the user input peaks and background features referring to the complementary regions. The four peak callers create the other four feature spaces, including the peak calling results as foreground features and the complementary regions as background features, respectively. Second, we plot the ROC curve and precision-recall curve by changing a threshold value across the p-values of peaks identified by each peak caller. In this comparison, we loosen the discovery rules when deploying each peak calling method to ensure that we have an enlarged list of discoveries. Third, we directly compare the peak calling results by counting the number of peaks identified simultaneously by multiple peak callers. We include the Venn diagram and upset plot output by Python package Intervene[47] to visualize the comparison results. The results are summarized in Fig. 6.

By examining the RPKM values for each peak calling method (Fig. 6A), we find that MACS3

has the best performance with the most similar pattern to the ground truth. SEACR has the lowest mean RPKM value for the foreground features among all four peak calling methods. One possible reason could be that SEACR tends to call peaks with a significantly longer length than other peak callers. In addition, the peak regions identified by SEACR might not include excessive biological signals as a natural chromatin open region does. Hence SEACR's average read coverage per base call might be lower than other methods. We also find that SEACR identifies more peaks than other methods: SEACR identifies 32768 peaks while the most peaks called by the other three methods only have thousands of peaks (Fig. 6D). This observation helps explain why in Fig. 6B, SEACR has a relatively low precision rate compared with others, as most of its peaks are not true positives. In addition, we find that MACS3 has the best ROC curve among the four peak callers but still leads to a relatively low precision when the recall is high, similar to the results in Figs. 3B, S1B. We notice that HMMRATAC, a peak caller specifically designed for ATAC-seq, performs well by having the highest precision while varying TPR. However, HMMRATAC finds the fewest peaks compared with other peak callers with the default setting.

Application 2: scReadSim is flexible for allowing varying throughputs and library sizes as input parameters to guide experimental design

In the synthetic count matrix generation procedure, scReadSim incorporates scDesign2[23] to train the statistical generative model and simulate the synthetic count matrix. The flexibility of scDesgin2 enables scReadSim to vary throughputs and library sizes. The mechanism is that scDesign2 models a joint distribution among all features and treats cells as independent observations following this parametric distribution. When changing the sequencing depth for output synthetic data, we could modify the joint distributions' mean parameters for features. Similarly, we can independently generate a certain amount of cells when specifying the throughput.

To illustrate this, we utilize the 10x scRNA-seq dataset (cell number 4877; sequencing depth around $20M$) and generate synthetic datasets with different amounts of cells or sequencing depths. Specifically, we fix the cell number and generate synthetic datasets with sequencing depth $2M(\times 0.125)$, $5M(\times 0.25)$, $10M(\times 0.5)$, $20M$(original amount), $40M(\times 2)$, $82M(\times 4)$, $164M(\times 8)$. Then by fixing the sequencing depth parameter, we generate synthetic datasets with cell number $609(\times 0.125)$, $1219(\times 0.25)$, $2438(\times 0.5)$, 4877(original amount), $9754(\times 2)$, $19508(\times 4)$, $39016(\times 8)$. For a clearer comparison, we visualize these BAM files through the IGV genome browser (Fig. S9). From the genome browser, we can observe a significantly increasing read coverage when the number of cells or the amount of sequencing depth increases (Fig. S9). This observation is within expectation because the mean of read counts sampled for each feature is larger when increasing the sequencing depth, and more cells are sampled from the joint distribution of features when specifying a large throughput. Due to the diploid nature of DNA and the sequencing mechanism of scATAC-seq, it is unlikely to have a limitless sequencing depth when preparing the scATAC-seq library. Thus we do not suggest users increase throughputs for the synthetic scATAC-seq reads.

Discussion

In this article, we propose scReadSim, a single-cell multi-omics read simulator. Trained on real data, scReadSim generates synthetic sequencing reads in BAM or FASTQ formats. There are three aspects that scReadSim differs from the currently existing simulators:

1. First, scReadSim deals with multiple omics data and achieves homogeneously good results when comparing the synthetic and real data in both read and count levels.

2. Second, scReadSim generates sequencing reads instead of a count matrix. The output synthetic reads support FASTQ and BAM formats, which are both standard inputs for low-level bioinformatics computational tools. While most single-cell sequencing data simulators only deal with the processed count matrix, limiting their usage solely to benchmarking count-level tools.

3. Third, scReadSim allows user-specified ground truths of accessible chromatin regions for single-cell chromatin accessibility data generation. In addition, scReadSim is flexible for allowing varying throughputs and library sizes as input parameters to guide experimental design.

We deploy scReadSim onto the sci-ATAC-seq dataset [41] and a single-cell multimodal dataset (10x Multiome dataset including both single-cell gene expression data and chromatin accessibility data) to verify that the synthetic data resembles the real data in both read and count levels, see Figs. 3, 5 and Supplementary Figs. S1, S3, S4, S5. In addition, we also apply a peak calling method MACS3 onto both real and synthetic BAM files to further verify that scReadSim indeed preserves the chromatin open information for the sequencing reads (Figs. 4, S2).

As a direct usage of a sequencing data simulator, one could evaluate the performance of computation tools using the simulated sequencing data. Our simulator, scReadSim, allows users to design the chromatin open regions for their synthetic scATAC-seq reads. We show that the synthetic reads' coverage resembles the read data regarding the peak regions and non-peak regions (Figs. S7, S8). Here the peaks for synthetic reads are input by users, while the peaks for real reads are decided using peak caller MACS3. This observation implies that scReadSim is able to preserve the sequence information from real data for the user-designed scATAC-seq synthetic data. We also include a benchmark study of peak calling methods for scATAC-seq data with manually created peudo-peaks: we truncate the gene transcript regions from TSS with random lengths as the peak regions. By comparing the peak calling results and the ground truth, input peaks, we find that the most popular peak caller, MACS3 has the best ROC curve. In addition, the HMMRATAC, the ATAC-seq peak caller, has the highest precision among all peak calling tools included in our study (Fig. 6).

Methods

Pre-process

Feature space

With a BAM file and its corresponding cell barcode information input by the user, scReadSim first identifies a feature space to construct the read count matrices with rows referring to features and columns representing cells. To define a feature space, scReadSim separates the reference genome into two categories of features: foreground features. Foreground features are those genome regions where the input BAM file contains more excessive biological variations than complementary regions. For example, scReadSim uses the peak calling method MACS3 to identify open chromatin regions as the foreground features for scATAC-seq and utilizes gene transcript regions as the foreground features for scRNA-seq data. We denote the corresponding complementary regions as the background features.

Existing count-level simulators for scATAC-seq implicitly fix the feature space when analyzing the real data. The simulated count entries correspond to the pre-defined features used in the real count matrix, like peaks or bin regions for scATAC-seq and genes for scRNA-seq. However, scReadSim allows users to specify the peak regions for their synthetic scATAC-seq reads arbitrarily. The mechanism is that scReadSim takes the user input peaks as the foreground features and treats the complementary regions as the corresponding background features. Meanwhile, scReadSim would identify a real feature space directly from the real data by calling peaks using MACS3. The foreground features of the real feature space are peaks identified from the real BAM file, and the background features are the corresponding complementary regions. Based on these two different feature spaces, scReadSim will summarize the real biological signals through the real feature space and then project it into the feature space defined by users to simulate scATAC-seq reads with self-defined peaks.

Count matrix construction

scReadSim counts the number of reads overlapped within each feature for every cell to construct the feature by cell barcode matrix. By default, scReadSim creates two count matrices regarding foreground and background features and treats two matrices separately in the following simulation procedure.

In case users specify peak regions for the synthetic scATAC-seq reads, scReadSim starts from two feature spaces: one feature space defined by users and the real feature space identified directly from the real data. Then scReadSim constructs two surjective mappings from the real feature space to the user-defined feature space based on the features' length similarity (Fig. 2B). In addition, scReadSim defines the mappings separately for foreground and background features, which means that a real foreground feature can only map into a user-input foreground feature. After

the mappings are defined, scReadSim treats the real feature's read counts as the mapped user-input feature's read counts, see Fig. 2C. In this way, we construct two pseudo-count matrices for the user-input foreground and background features. Then the constructed pseudo-count matrices would be used for model fitting, synthetic count matrix generation, and synthetic reads sampling in the following steps to generate the user-designed scATAC-seq synthetic data(Fig. 2A).

Synthetic count matrix generation

The synthetic count matrix generation can essentially utilize any count-level single-cell data simulation algorithms. The scRNA-seq count simulator scDesign2[23] outperforms other existing simulators by capturing the correlation of the genes through a copula structure. Hence to ensure a homogeneously good performance modeling reads count levels across different omics, scReadSim incorporates scDesign2 with minor modifications to generate the synthetic count matrix. The procedure mainly involves two steps: model fitting based on the real input matrix and synthetic count matrix generation based on the fitted probabilistic model. Note that scDesign2 trains models and simulates synthetic data independently for the pre-defined cell types or clusters. As pointed out by scDesign2's authors, the partitioning of cell types, or cell clustering, is an essential pre-processing step of scDesign2. Hence to ensure the performance of scDesign2, scReadSim provides a built-in clustering step before deploying scDesign2 when the users do not provide the cell labels. We use the Louvain clustering function of Seurat[48] package. One of the output files of scReadSim would be the cell labels corresponding to each synthetic cell.

Synthetic reads generation based on synthetic count data

Given the generated synthetic count matrix, scReadSim directly samples synthetic reads from the input real BAM file. For each feature, scReadSim first samples without replacement the particular number of real reads overlapped with that region from the real BAM file. The sampled read amount is decided according to the synthetic count matrix for each feature and cell. When sampling reads, scReadSim records the reads' starting positions along the reference genome. For paired-end reads, scReadSim records each real read's starting position, mate read's starting position, and the fragment length when sampling the reads. Given the read length specified by users, scReadSim decides the ending position for each sampled read and extracts the sequencing base calls from the reference genome according to the read's starting and ending positions. These sampled reads are then formatted into a FASTQ file with a perfect sequencing quality for each base call or further aligned onto the reference genome to create a BAM file for output. For paired-end reads sequencing technologies, scReadSim samples read pairs for each feature. Specifically, we first decide the number of read pairs sampled for one feature by simply dividing the synthetic count matrix by 2. Then scReadSim samples this amount of reads from real BAM file and also extracts the mate reads if one mate of the read pair is sampled out.

The polymerase chain reaction (PCR) amplification ensures sufficient DNA fragments for se-

quencing but may lead to biases that some fragments are excessively present in the final library[49]. Hence, researchers usually remove duplicate reads to avoid pseudo-signals created by PCR amplification when preparing the raw data[50]. To avoid duplicates when bootstrapping the reads from real data, scReadSim adds a random shift when recording the starting positions of each sampled read. The number of base pairs shifted is randomly chosen from the set $\{-4, -3, \ldots, 3, 4\}$, which could be specified by users. For paired-end sequencing technologies, to avoid duplicate reads, scReadSim also introduces a random noise to both mates' starting positions of a read pair, which keeps a constant fragment length even after the random shift.

When users input peak regions for synthetic scATAC-seq data, scReadSim will sample reads based on the mapping between the feature space defined by users and the real feature space. Specifically, for each synthetic cell, scReadSim samples a certain number of reads for each user-defined feature from the real reads overlapped with its corresponding real feature. Then the sampled reads' starting positions would be adjusted based on the distance between the real and user-input features. The simulated count matrix decides the number of reads to be sampled. scReadSim also adds a random shift to the starting position of each synthetic read to avoid technical duplicates.

Generation of synthetic FASTQ and BAM files

From the sampled synthetic reads to the FASTQ or BAM file for output, scReadSim first keeps the coordinates of synthetic reads within a bed file. Then use bedtools[51] `getfasta` function to extract the corresponding sequences from the reference genome for every synthetic read and output in a FASTQ format using seqtk[52] software. For paired-end reads, scReadSim would generate two FASTQ files for the two mates of read pairs separately. Users could choose FASTQ files as the output of scReadSim, which are the input of most analysis pipelines for single-cell sequencing data. In addition, scReadSim could output synthetic reads in BAM files by aligning the reads onto the specified reference genome using alignment software bowtie2[34].

Substitution errors introduced into synthetic reads

Generating synthetic reads by slicing reference genome into pieces will guarantee each base call of the reads with perfect sequencing qualities. To mimic the error rates observed in the actual Illumina sequencing reads, scReadSim supports introducing substitution errors into synthetic reads. To do so, scReadSim first incorporates fgbio[53] to calculate the error rate per base call for real reads. Then for the perfect-quality synthetic reads, scReadSim decides if every base call is erroneous according to the real reads' error rates. If one base call is decided to be erroneous, the substitution error from one nucleotide into the other three nucleotides is randomly sampled from a multi-categorical distribution with probabilities in proportion to the real substitution errors in the same position. As the two mates of read pairs from paired-end sequencing technologies have different error rates per base, scReadSim generates substitution errors separately for the two

mates of synthetic read pairs.

Evaluation metrics

K-mer spectrum We use tool Jellyfish[54] to obtain the 11-mer, 21-mer, 31-mer, 41-mer spectrum for both synthetic and real BAM files with the default settings.
Fragment length distribution Fragment lengths for paired-end sequencing data are obtained using function `bamPEFragmentSize` of the software deeptools[55] with the default settings.
Error rates The substitution error rates per base within reads are obtained using software fgbio[53] with function `ErrorRateByReadPosition` by setting option collapse as false.
Reads Per Kilobase Million (RPKM) The RPKM values for a given set of features are defined as

$$\text{RPKM} = \frac{\text{count per feature} \times 1,000,000}{\text{total count of reads} \times \text{feature length in kilobase}}.$$

miLISI The integration local inverse Simpson's Index (iLISI)[46] for each cell measures the effective number of datasets in a neighborhood of the cell. A cell surrounded by a single dataset has a iLISI value equal to 1, while if it has a neighborhood with an equal number of cells from two datasets, the iLISI value would be 2. Hence the median iLISI (miLISI) measures the average level of datasets mixture, and a miLISI close to 2 means that two datasets have a perfect mixing pattern within the embedding.

Availability of data and materials

Datasets

- **10x Genomics**: The 10x Genomics datasets include single-cell Multiome ATAC and Gene Expression data, which measure the embryonic mouse brain tissue. The raw BAM files are downloaded from Fresh embryonic E18 mouse brain (5k).

- **sci-ATAC-seq**: The sci-ATAC-seq dataset measures the chromatin accessibility for 17 samples spanning 13 different tissues in 8-week old mice[41]. The raw BAM file is downloaded from `https://atlas.gs.washington.edu/mouse-atac/`. Data for tissue Bone Marrow 62016 is selected for analysis.

Software and code

The `scReadSim` Python package is available at `https://github.com/Dominic7227/scReadSim`. The source code and data for reproducing the results are available at fill in. In addition, this work makes use of the following tools:

- **fgbio**: Software fgbio Version 2.0.1 is used for the analysis (`http://fulcrumgenomics.github.io/fgbio/`).
- **Jellyfish**: Software Jellyfish Version 2.3.0 is used for the analysis (`https://github.com/gmarcais/Jellyfish`).
- **deeptools**: Software deeptools Version 3.5.0 is used for the analysis (`https://deeptools.readthedocs.io/en/develop/content/tools/bamPEFragmentSize.html`).
- **Intervene**: Python package Intervene Version 0.41.0 is used for the analysis (`https://intervene.readthedocs.io/en/latest/install.html`).

Dimension reduction methods

- **UMAP**: Function `umap` from R package umap Version 0.2.7.0 is used for the analysis.
- **PCA**: Function `irlba` from R package irlba 2.3.5 is used for the analysis.

Peak calling methods

- **MACS3**: Software MACS3 Version 3.0.0a7 is used for the analysis (`https://github.com/macs3-project/MACS`).
- **SEACR**: Software SEACR Version 1.3 is used for the analysis (`https://github.com/FredHutch/SEACR`).
- **HOMER**: Software HOMER Version 4.11 is used for the analysis (`http://homer.ucsd.edu/homer/index.html`).
- **HMMRATAC**: Software HMMRATAC Version 1.2.10 is used for the analysis (`https://github.com/LiuLabUB/HMMRATAC`).

Essential tools of scReadSim

The following tools along with MACS3[29] are essential tools when implementing scReadSim. We report the tools' versions used when developing scReadSim.

- **samtools**: Software samtools Version samtools 1.12 (with htslib 1.12) is used for the analysis (`http://www.htslib.org/`).
- **bedtools**: Software bedtools Version 2.29.1 is used for the analysis (`https://bedtools.readthedocs.io/en/latest/`).
- **seqtk**: Software seqtk Version 1.3-r117-dirty is used for the analysis (`https://github.com/lh3/seqtk`).

- **bowtie2**: Software fgbio Version 2.3.4.1 is used for the analysis (`http://bowtie-bio.sourceforge.net/bowtie2/index.shtml`).
- **Seurat**: R pacakge Seurat Version 4.0.6 is used for the analysis.

Competing interests

The authors declare no competing interests.

Acknowledgements

The authors appreciate the comments and feedback from the members of the Junction of Statistics and Biology at UCLA (`http://jsb.ucla.edu`).

Funding

This work was supported by the following grants: National Science Foundation DBI-1846216 and DMS-2113754, NIH/NIGMS R01GM120507 and R35GM140888, Johnson Johnson WiSTEM2D Award, Sloan Research Fellowship, and UCLA David Geffen School of Medicine W.M. Keck Foundation Junior Faculty Award (to J.J.L.).

References

[1] Yukie Kashima, Yoshitaka Sakamoto, Keiya Kaneko, Masahide Seki, Yutaka Suzuki, and Ayako Suzuki. Single-cell sequencing techniques from individual to multiomics analyses. *Experimental & Molecular Medicine*, 52(9):1419–1427, 2020.

[2] Hongshan Guo, Ping Zhu, Xinglong Wu, Xianlong Li, Lu Wen, and Fuchou Tang. Single-cell methylome landscapes of mouse embryonic stem cells and early embryos analyzed using reduced representation bisulfite sequencing. *Genome Research*, 23(12):2126–2135, 2013.

[3] Sébastien A. Smallwood, Heather J. Lee, Christof Angermueller, Felix Krueger, Heba Saadeh, Julian Peat, Simon R. Andrews, Oliver Stegle, Wolf Reik, and Gavin Kelsey. Single-cell genome-wide bisulfite sequencing for assessing epigenetic heterogeneity. *Nature Methods*, 11(8):817–820, 2014.

[4] Assaf Rotem, Oren Ram, Noam Shoresh, Ralph A. Sperling, Alon Goren, David A. Weitz, and Bradley E. Bernstein. Single-cell ChIP-seq reveals cell subpopulations defined by chromatin state. *Nature Biotechnology*, 33(11):1165–1172, 2015.

[5] Kevin Grosselin, Adeline Durand, Justine Marsolier, Adeline Poitou, Elisabetta Marangoni, Fariba Nemati, Ahmed Dahmani, Sonia Lameiras, Fabien Reyal, Olivia Frenoy, Yannick Pousse, Marcel Reichen, Adam Woolfe, Colin Brenan, Andrew D. Griffiths, Céline Vallot, and Annabelle Gérard. High-throughput single-cell ChIP-seq identifies heterogeneity of chromatin states in breast cancer. *Nature Genetics*, 51(6):1060–1066, 2019.

[6] Steven J. Wu, Scott N. Furlan, Anca B. Mihalas, Hatice S. Kaya-Okur, Abdullah H. Feroze, Samuel N. Emerson, Ye Zheng, Kalee Carson, Patrick J. Cimino, C. Dirk Keene, Jay F. Sarthy, Raphael Gottardo, Kami Ahmad, Steven Henikoff, and Anoop P. Patel. Single-cell CUT&Tag analysis of chromatin modifications in differentiation and tumor progression. *Nature Biotechnology*, 2021.

[7] Hatice S. Kaya-Okur, Steven J. Wu, Christine A. Codomo, Erica S. Pledger, Terri D. Bryson, Jorja G. Henikoff, Kami Ahmad, and Steven Henikoff. CUT&Tag for efficient epigenomic profiling of small samples and single cells. *Nature Communications*, 10(1):1930, 2019.

[8] Jason D. Buenrostro, Beijing Wu, Ulrike M. Litzenburger, Dave Ruff, Michael L. Gonzales, Michael P. Snyder, Howard Y. Chang, and William J. Greenleaf. Single-cell chromatin accessibility reveals principles of regulatory variation. *Nature*, 523(7561):486–490, 2015.

[9] Darren A. Cusanovich, Riza Daza, Andrew Adey, Hannah A. Pliner, Lena Christiansen, Kevin L. Gunderson, Frank J. Steemers, Cole Trapnell, and Jay Shendure. Multiplex single cell profiling of chromatin accessibility by combinatorial cellular indexing. *Science (New York, N.Y.)*, 348(6237):910–914, 2015.

[10] Takashi Nagano, Yaniv Lubling, Tim J. Stevens, Stefan Schoenfelder, Eitan Yaffe, Wendy Dean, Ernest D. Laue, Amos Tanay, and Peter Fraser. Single-cell Hi-C reveals cell-to-cell variability in chromosome structure. *Nature*, 502(7469):59–64, 2013.

[11] Ashraful Haque, Jessica Engel, Sarah A Teichmann, and Tapio Lönnberg. A practical guide to single-cell rna-sequencing for biomedical research and clinical applications. *Genome medicine*, 9(1):1–12, 2017.

[12] Qiaolin Deng, Daniel Ramsköld, Björn Reinius, and Rickard Sandberg. Single-cell rna-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, 343 (6167):193–196, 2014.

[13] Alex K Shalek, Rahul Satija, Xian Adiconis, Rona S Gertner, Jellert T Gaublomme, Raktima Raychowdhury, Schraga Schwartz, Nir Yosef, Christine Malboeuf, Diana Lu, et al. Single-cell transcriptomics reveals bimodality in expression and splicing in immune cells. *Nature*, 498 (7453):236–240, 2013.

[14] Philip Brennecke, Simon Anders, Jong Kyoung Kim, Aleksandra A Kołodziejczyk, Xiuwei Zhang, Valentina Proserpio, Bianka Baying, Vladimir Benes, Sarah A Teichmann, John C Marioni, et al. Accounting for technical noise in single-cell rna-seq experiments. *Nature methods*, 10(11):1093–1095, 2013.

[15] Luke Zappia and Fabian J Theis. Over 1000 tools reveal trends in the single-cell rna-seq analysis landscape. *Genome biology*, 22(1):1–18, 2021.

[16] Xian Fan Mallory and Luay Nakhleh. SimSCSnTree: A simulator of single-cell DNA sequencing data. *Bioinformatics*, 2022.

[17] Collin Giguere, Harsh Vardhan Dubey, Vishal Kumar Sarsani, Hachem Saddiki, Shai He, and Patrick Flaherty. SCSIM: Jointly simulating correlated single-cell and bulk next-generation DNA sequencing data. *BMC Bioinformatics*, 21(1):1–10, 2020.

[18] Zhenhua Yu, Fang Du, Xuehong Sun, and Ao Li. SCSsim: An integrated tool for simulating single-cell genome sequencing data. *Bioinformatics*, 36(4):1281–1282, 2020.

[19] David Posada and Jeffrey Townsend. CellCoal: Coalescent Simulation of Single-Cell Sequencing Samples. *Molecular Biology and Evolution*, 37(5):1535–1542, 2020.

[20] Hamim Zafar, Nicholas Navin, Luay Nakhleh, and Ken Chen. Computational approaches for inferring tumor evolution from single-cell genomic data. *Current Opinion in Systems Biology*, 7:16–25, 2018.

[21] Xian F Mallory, Mohammadamin Edrisi, Nicholas Navin, and Luay Nakhleh. Assessing the performance of methods for copy number aberration detection from single-cell DNA sequencing data. *PLoS computational biology*, 16(7):e1008012, 2020.

[22] Wei Vivian Li and Jingyi Jessica Li. A statistical simulator scdesign for rational scrna-seq experimental design. *Bioinformatics*, 35(14):i41–i50, 2019.

[23] Tianyi Sun, Dongyuan Song, Wei Vivian Li, and Jingyi Jessica Li. scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. *Genome biology*, 22(1):1–37, 2021.

[24] Luke Zappia, Belinda Phipson, and Alicia Oshlack. Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):1–15, 2017.

[25] Davide Risso, Fanny Perraudeau, Svetlana Gribkova, Sandrine Dudoit, and Jean-Philippe Vert. A general and flexible method for signal extraction from single-cell rna-seq data. *Nature communications*, 9(1):1–17, 2018.

[26] Zeinab Navidi, Lin Zhang, and Bo Wang. simATAC: a single-cell ATAC-seq simulation framework. *Genome biology*, 22(1):1–16, 2021.

[27] Zhanlin Chen, Jing Zhang, Jason Liu, Zixuan Zhang, Jiangqi Zhu, Donghoon Lee, Min Xu, and Mark Gerstein. SCAN-ATAC-Sim: a scalable and efficient method for simulating single-cell ATAC-seq data from bulk-tissue experiments. *Bioinformatics*, 37(12):1756–1758, 2021.

[28] Wenshan Li and Wei Wang. Read simulator for single cell RNA sequencing. *UCLA Electronic Theses and Dissertations*, 2019.

[29] Yong Zhang, Tao Liu, Clifford A Meyer, Jérôme Eeckhoute, David S Johnson, Bradley E Bernstein, Chad Nusbaum, Richard M Myers, Myles Brown, Wei Li, et al. Model-based analysis of chip-seq (macs). *Genome biology*, 9(9):1–9, 2008.

[30] Sven Heinz, Christopher Benner, Nathanael Spann, Eric Bertolino, Yin C Lin, Peter Laslo, Jason X Cheng, Cornelis Murre, Harinder Singh, and Christopher K Glass. Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and b cell identities. *Molecular cell*, 38(4):576–589, 2010.

[31] Evan D Tarbell and Tao Liu. HMMRATAC: a Hidden Markov ModeleR for ATAC-seq. *Nucleic acids research*, 47(16):e91–e91, 2019.

[32] Michael P Meers, Dan Tenenbaum, and Steven Henikoff. Peak calling by sparse enrichment analysis for cut&run chromatin profiling. *Epigenetics & chromatin*, 12(1):1–11, 2019.

[33] Reuben Thomas, Sean Thomas, Alisha K Holloway, and Katherine S Pollard. Features that define the best chip-seq peak calling algorithms. *Briefings in bioinformatics*, 18(3):441–450, 2017.

[34] Ben Langmead and Steven L Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, 2012.

[35] Rongxin Fang, Sebastian Preissl, Yang Li, Xiaomeng Hou, Jacinta Lucero, Xinxin Wang, Amir Motamedi, Andrew K Shiau, Xinzhu Zhou, Fangming Xie, et al. Comprehensive analysis of single cell atac-seq data with snapatac. *Nature communications*, 12(1):1–15, 2021.

[36] Youjin Hu, Qin An, Katherine Sheu, Brandon Trejo, Shuxin Fan, and Ying Guo. Single cell multi-omics technology: methodology and application. *Frontiers in cell and developmental biology*, 6:28, 2018.

[37] Anjun Ma, Adam McDermaid, Jennifer Xu, Yuzhou Chang, and Qin Ma. Integrative methods and practical challenges for single-cell multi-omics. *Trends in biotechnology*, 38(9):1007–1022, 2020.

[38] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M Mauck III, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive integration of single-cell data. *Cell*, 177(7):1888–1902, 2019.

[39] Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular systems biology*, 14(6):e8124, 2018.

[40] Robrecht Cannoodt, Wouter Saelens, Louise Deconinck, and Yvan Saeys. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications*, 12(1):1–9, 2021.

[41] Darren A Cusanovich, Andrew J Hill, Delasa Aghamirzaie, Riza M Daza, Hannah A Pliner, Joel B Berletch, Galina N Filippova, Xingfan Huang, Lena Christiansen, William S DeWitt, et al. A single-cell atlas of in vivo mammalian chromatin accessibility. *Cell*, 174(5):1309–1324, 2018.

[42] James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. Integrative genomics viewer. *Nature biotechnology*, 29(1): 24–26, 2011.

[43] Stephan Schmeing and Mark D. Robinson. ReSeq simulates realistic Illumina high-throughput sequencing data. *Genome Biology*, 22(1):1–37, 2021.

[44] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[45] Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 37(1):38–44, 2019.

[46] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with harmony. *Nature methods*, 16(12):1289–1296, 2019.

[47] Tiziano Flati, Silvia Gioiosa, Giovanni Chillemi, Andrea Mele, Alberto Oliverio, Cecilia Mannironi, Arianna Rinaldi, and Tiziana Castrignanò. A gene expression atlas for different kinds of stress in the mouse brain. *Scientific Data*, 7(1):1–18, 2020.

[48] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck III, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[49] Vivien Marx. How to deduplicate PCR. *Nature Methods*, 14(5):473–476, 2017.

[50] Tom Smith, Andreas Heger, and Ian Sudbery. UMI-tools: Modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. *Genome Research*, 27(3): 491–499, 2017.

[51] Aaron R Quinlan and Ira M Hall. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

[52] Wei Shen, Shuai Le, Yan Li, and Fuquan Hu. Seqkit: a cross-platform and ultrafast toolkit for fasta/q file manipulation. *PloS one*, 11(10):e0163962, 2016.

[53] Tim Fennell and Nils Homer. fgbio. `https://github.com/fulcrumgenomics/fgbio`, 2019.

[54] G Marcais and C Kingsford. Jellyfish: A fast k-mer counter. *Tutorialis e Manuais*, 1:1–8, 2012.

[55] Fidel Ramírez, Devon P Ryan, Björn Grüning, Vivek Bhardwaj, Fabian Kilpert, Andreas S Richter, Steffen Heyne, Friederike Dündar, and Thomas Manke. deeptools2: a next generation web server for deep-sequencing data analysis. *Nucleic acids research*, 44(W1): W160–W165, 2016.

Figures

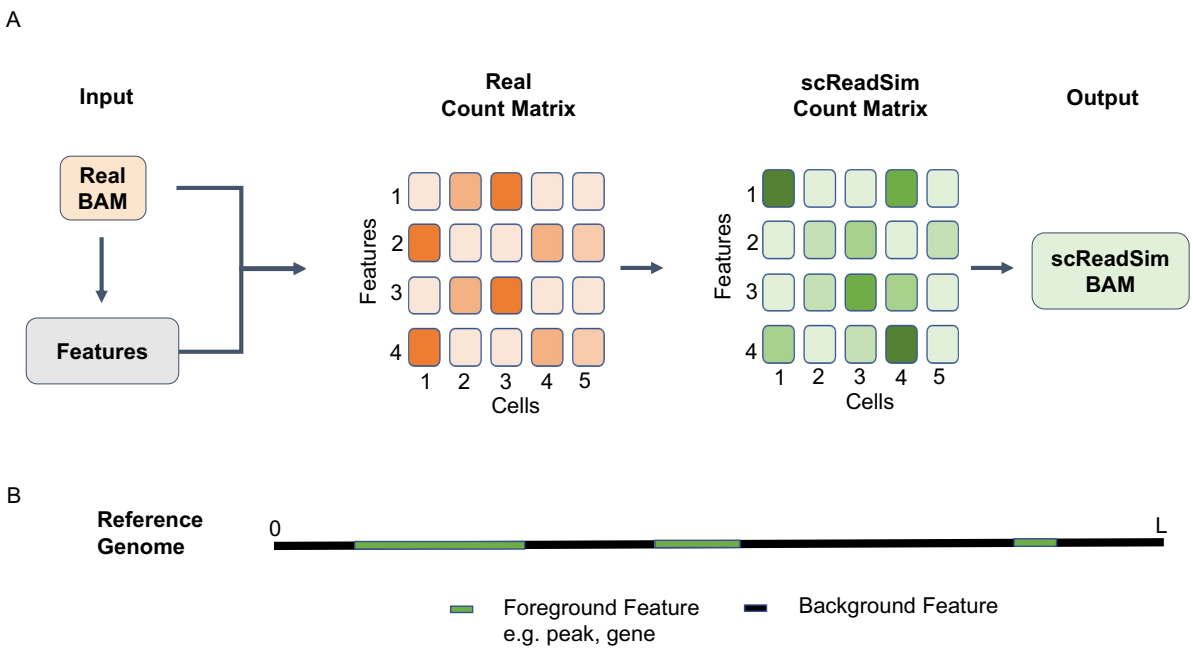

Figure 1: An overview of scReadSim. **A** scReadSim takes a BAM file as input and defines the feature space to convert the original BAM file into a count matrix. Based on the constructed count matrix, scReadSim trains a generative model and simulates a synthetic count matrix with the user-specified cell number and sequencing depth. The synthetic count matrix then guides scReadSim to sample reads from the real BAM file and convert the sampled synthetic reads into a BAM file for output. **B** The reference genome used to align the synthetic reads is split into the foreground and background features, which form the feature space used in the scReadSim analysis. Specifically, the regions where the real BAM file has excessive biological information are foreground features, like peaks for scATAC-seq and genes for scRNA-seq. The complementary regions of foreground features are background features.

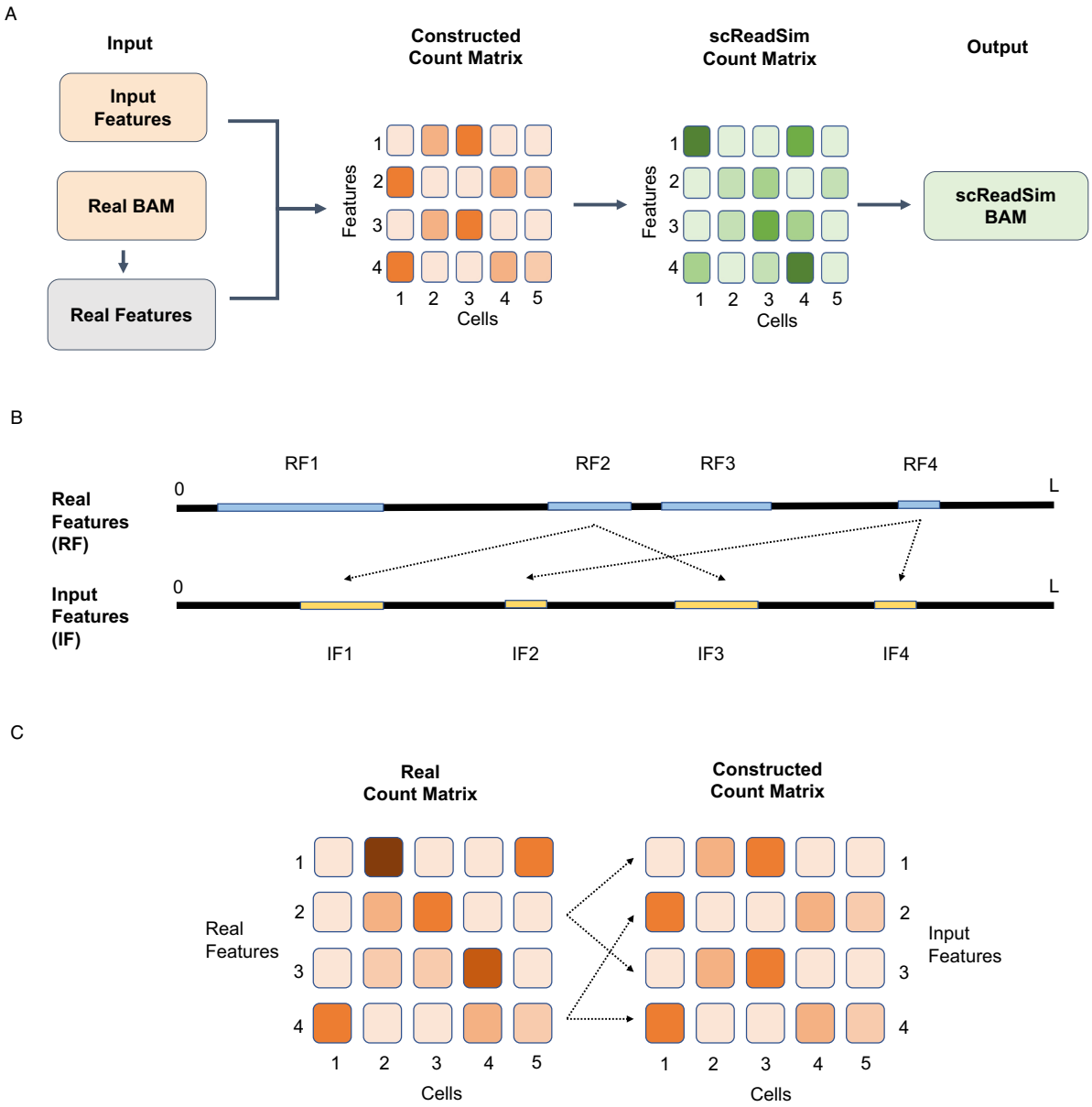

Figure 2: The pipeline of scReadSim with user-input feature sets. **A** scReadSim allows users to specify chromatin open regions (peaks) for their synthetic BAM file. Given the input real BAM file and peak regions, scReadSim uses two feature spaces: one is the feature space defined by users, and the other one is the real feature space identified directly from the real BAM file. Specifically, the feature space defined by users takes the user input peaks as the foreground features and the complementary regions as background features. scReadSim then constructs a count matrix with rows corresponding to the user input features based on a mapping between the two feature spaces (**B, C**). Like the case without user input features, scReadSim trains a generative model on the constructed count matrix and simulates a synthetic count matrix used for sampling reads for each user-specified feature. The sampled synthetic reads are stored and output as a BAM file. **B** The mappings from real features onto input features are defined based on the similarity of feature lengths. scReadSim assigns each input feature with a real feature if they have the closest feature-length. **C** The constructed count matrix for user input features is a matrix by concatenating vertically the real count vectors belonging to the real features mapped onto the user input features.

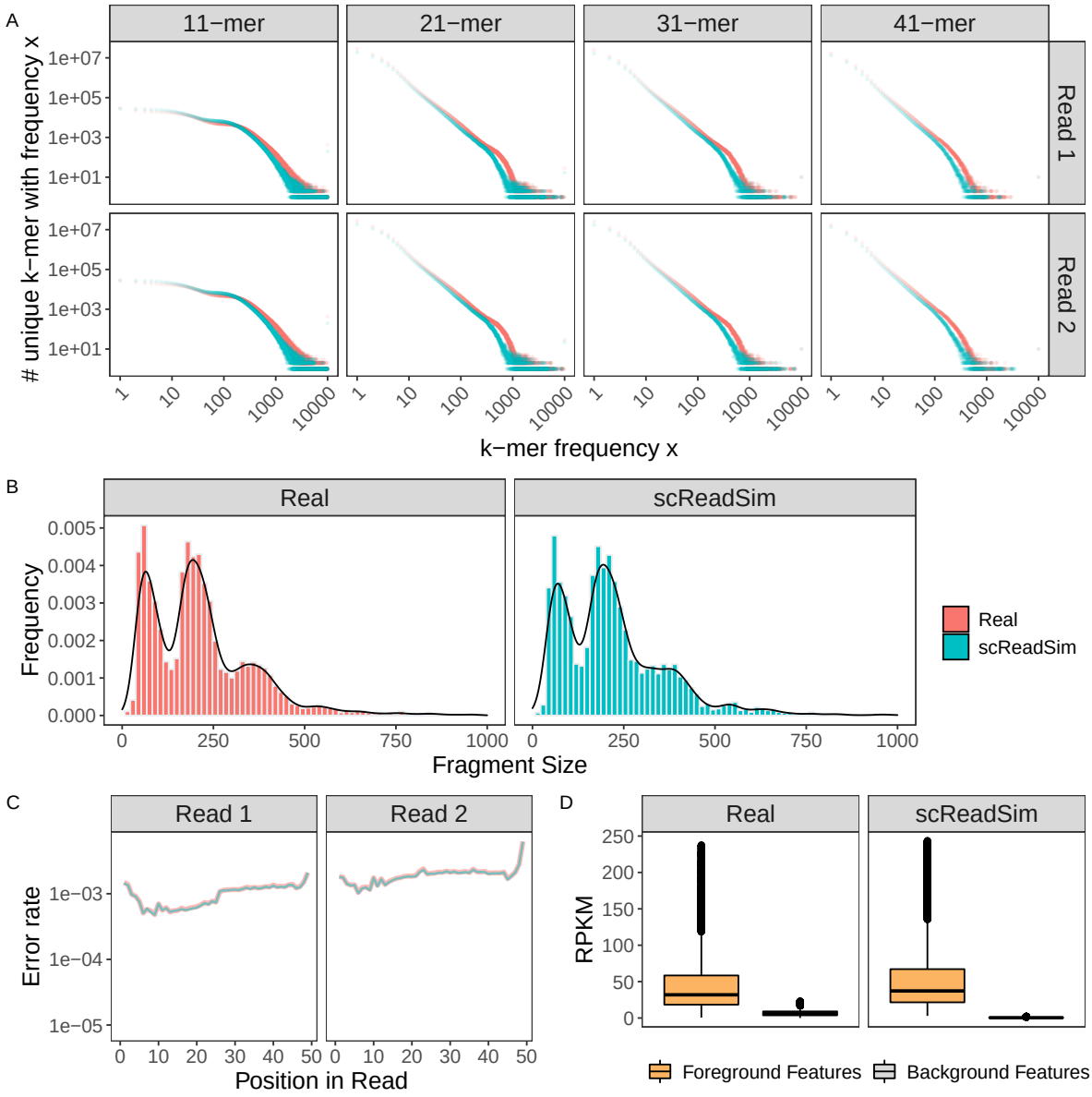

Figure 3: Synthetic reads generated by scReadSim resemble real 10x scATAC-seq data in read level. **A** K-mer spectrum comparison between scReadSim synthetic and real data. The x-axis refers to the frequency that occurred for a specific k-mer, while the y-axis represents the number of unique k-mers with this frequency. Both x-axis and y-axis are in log_{10} scale. **B** Fragment length distribution comparison between scReadSim synthetic and real data. **C** Error rate per base call within the read. The line plot's x-axis represents the positions of bases within a read, and the y-axis refers to the substitution error rates corresponding to that position. **D** Reads Per Kilobase Million (RPKM) of scReadSim synthetic and real BAM files. Features are categorized into the foreground and background features. The foreground features are peaks identified using MACS3 on the real BAM file, while the background features are the complementary regions of foreground features.

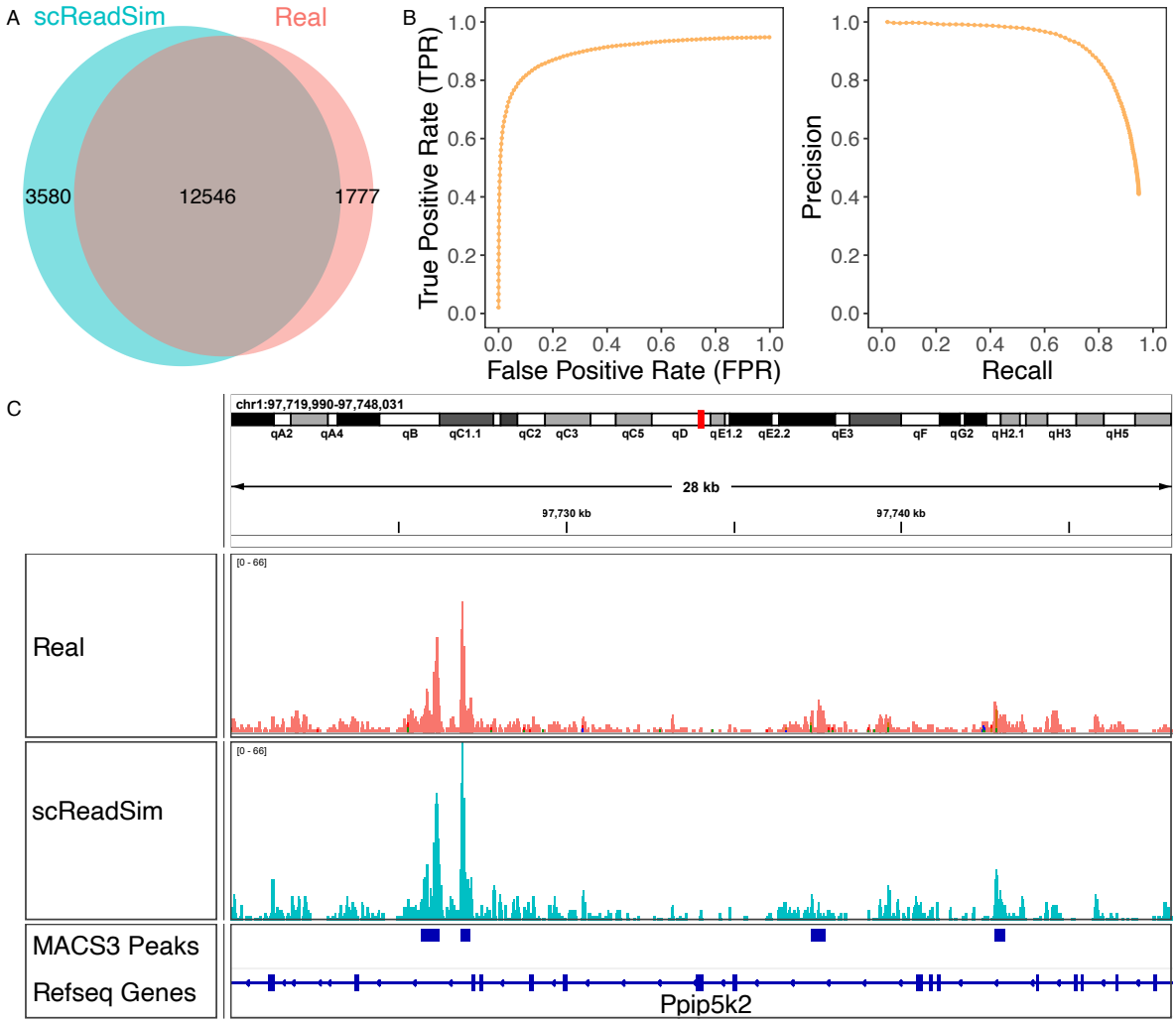

Figure 4: Synthetic reads generated by scReadSim resemble real 10x scATAC-seq data in terms of chromatin open regions. **A** Venn diagram of peak calling results using MACS3 on both scReadSim synthetic and real BAM files. **B** True positive rate (TPR) vs. false positive rate (FPR) and precision vs. recall by taking real peaks as truths. The peak identified using a synthetic BAM file is taken as a true peak if the overlapped length between it and one real peak is larger than half of the minimal scReadSim peak length. **C** Read coverage comparison through IGV genome browser. BAM files of the scReadSim synthetic data and real data are aligned side by side onto the reference genome. The peak regions called using MACS3 on real BAM file are shown in the MACS3 Peaks track.

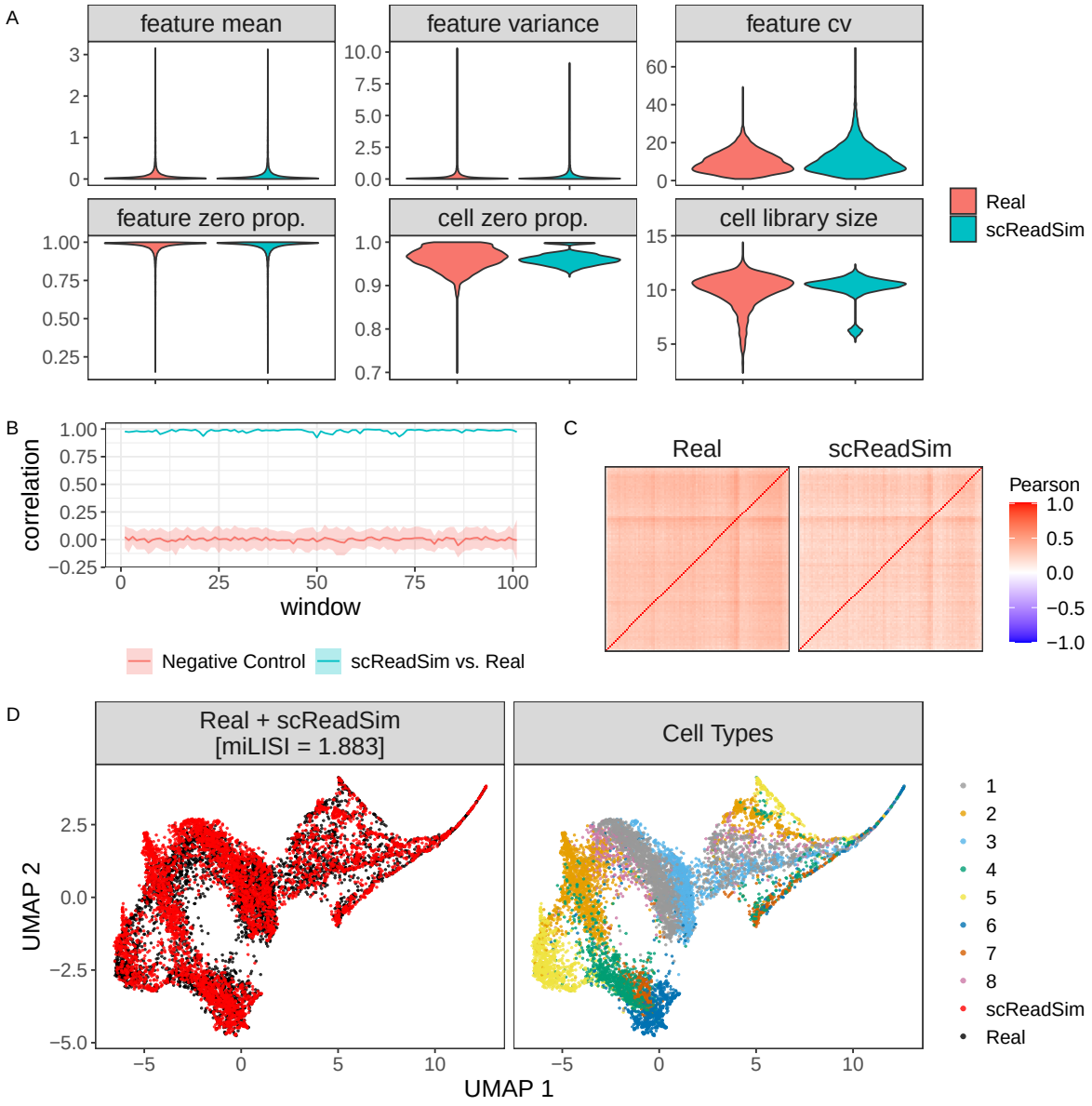

Figure 5: Synthetic reads generated by scReadSim resemble real 10x scATAC-seq data in count level. **A** Summary statistics of the count matrix for feature level (mean, variance, coefficient of variance (cv), zero proportion) and cell level (zero proportion and cell library size). **B** Correlation of read coverage between scReadSim synthetic and real BAM among 100 windows along the reference genome. The negative control group in red color represents the correlation of real BAM read coverage between random pairs of windows. Random sampling repeats 100 times for each window, and the mean and standard deviation are reported. **C**. Correlation among 100 top-expressed features using the synthetic and real count matrix. The top-expressed features are selected using the real count matrix. **D** UMAP visualization of the mixture of real and scReadSim synthetic count matrices. Cells in the left figure are labeled by the dataset source: real or synthetic cells. In the title of the left figure, miLISI measures the mixture of synthetic and real data points, which is close to 2 if they are perfectly mixed. Cells in the right figure are labeled by the cell clusters estimated during the synthetic count matrix generation process within scReadSim.

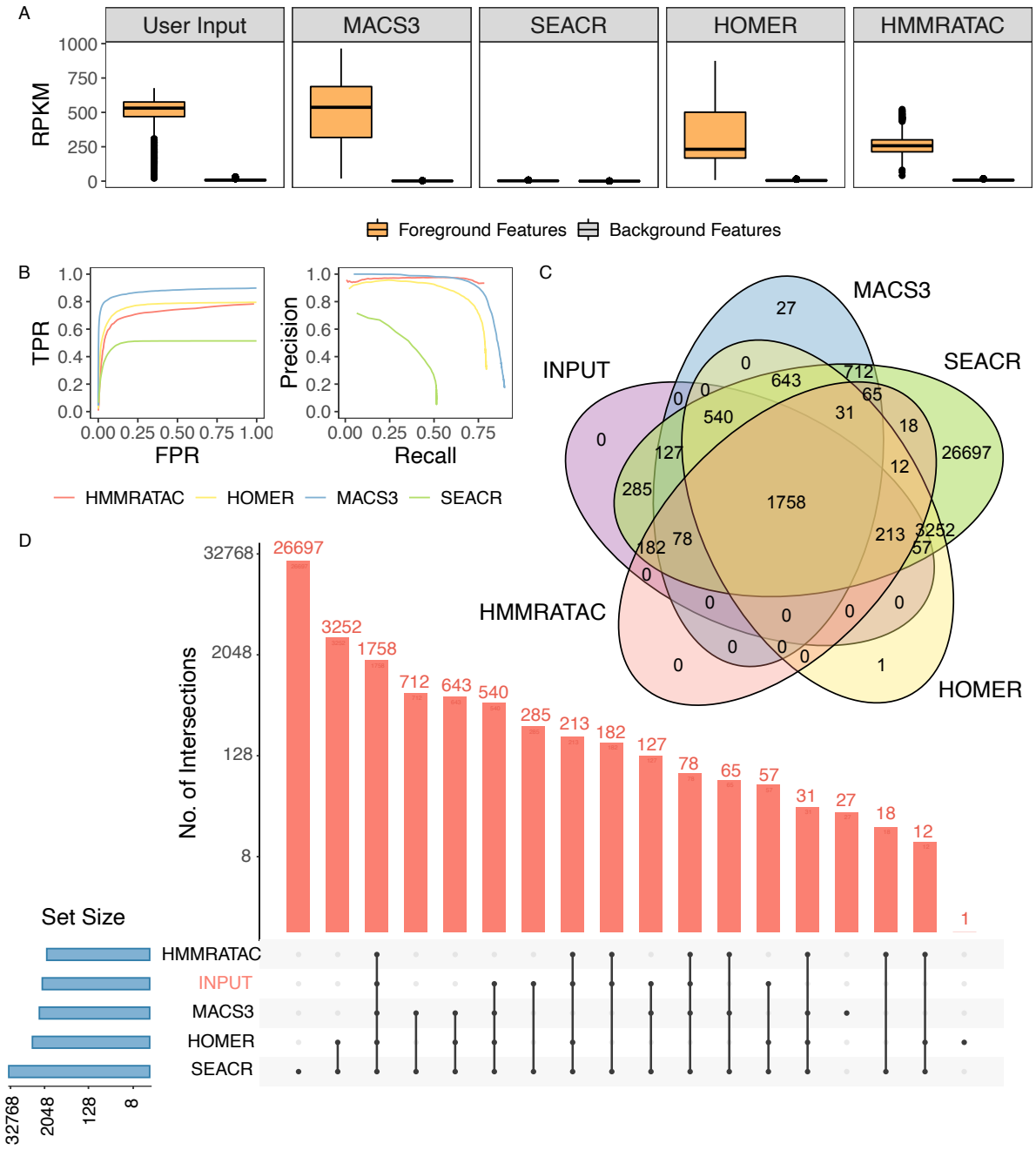

Figure 6: Benchmark study using synthetic reads generated by scReadSim with user-specified chromatin open regions for peak calling methods: MACS3, SEACR, HOMER, and HMMRATAC. **A** Reads Per Kilobase Million (RPKM) of scReadSim synthetic and read BAM files. Features are categorized into the foreground and background features. Foreground features for the User Input case refer to the input peaks, while for the four peak callers, the foreground features are peaks identified using each peak caller on the scReadSim synthetic BAM file. The background features are the complementary regions of the corresponding foreground features for each case. **B** True positive rate (TPR) vs. false positive rate (FPR) curve and precision vs. recall curve by taking input peaks as the truths for the four peak callers. One peak identified using the synthetic BAM file by a peak caller would be taken as a true peak if the overlapped length between the synthetic and one input peak is larger than half of the minimal scReadSim peak length. **C** Venn diagram of the input peak set and the four peak sets generated by the peak callers. **D** Upset plot of the input peak set and the four peak sets generated by the peak callers.

Supplementary Figures

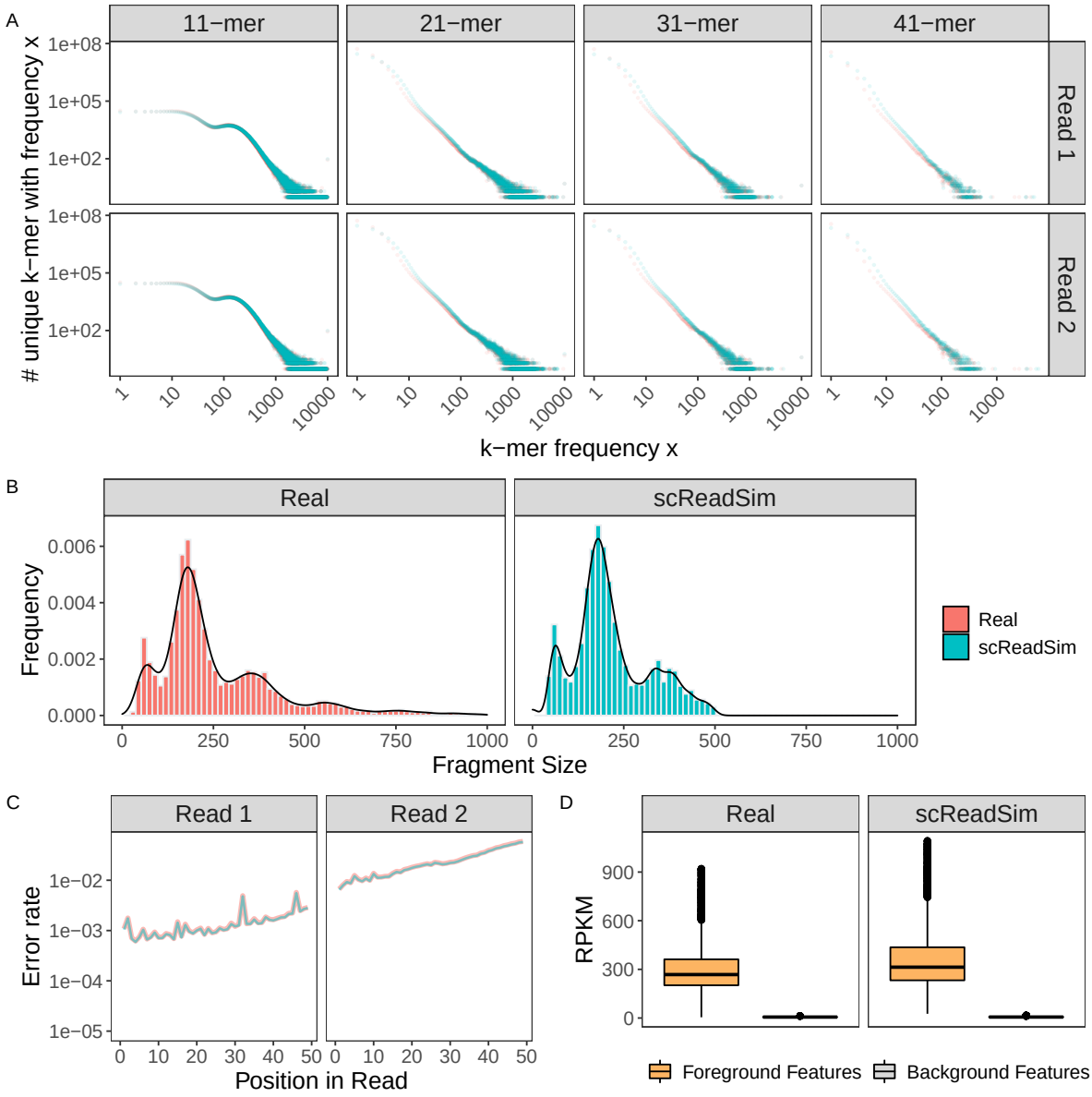

Figure S1: Synthetic reads generated by scReadSim resemble real sci-ATAC-seq data in read level. **A** K-mer spectrum comparison between scReadSim synthetic and real data. The x-axis refers to the frequency that occurred for a specific k-mer, while the y-axis represents the number of unique k-mers with this frequency. Both x-axis and y-axis are in log_{10} scale. **B** Fragment length distribution comparison between scReadSim synthetic and real data. **C** Error rate per base call within the read. The line plot's x-axis represents the positions of bases within a read and the y-axis refers the substitution error rates corresponding to that position. **D** Reads Per Kilobase Million (RPKM) of scReadSim synthetic and real BAM files. Features are categorized into the foreground and background features. The foreground features are peaks identified using MACS3 on the real BAM file, while the background features are the complementary regions of foreground features.

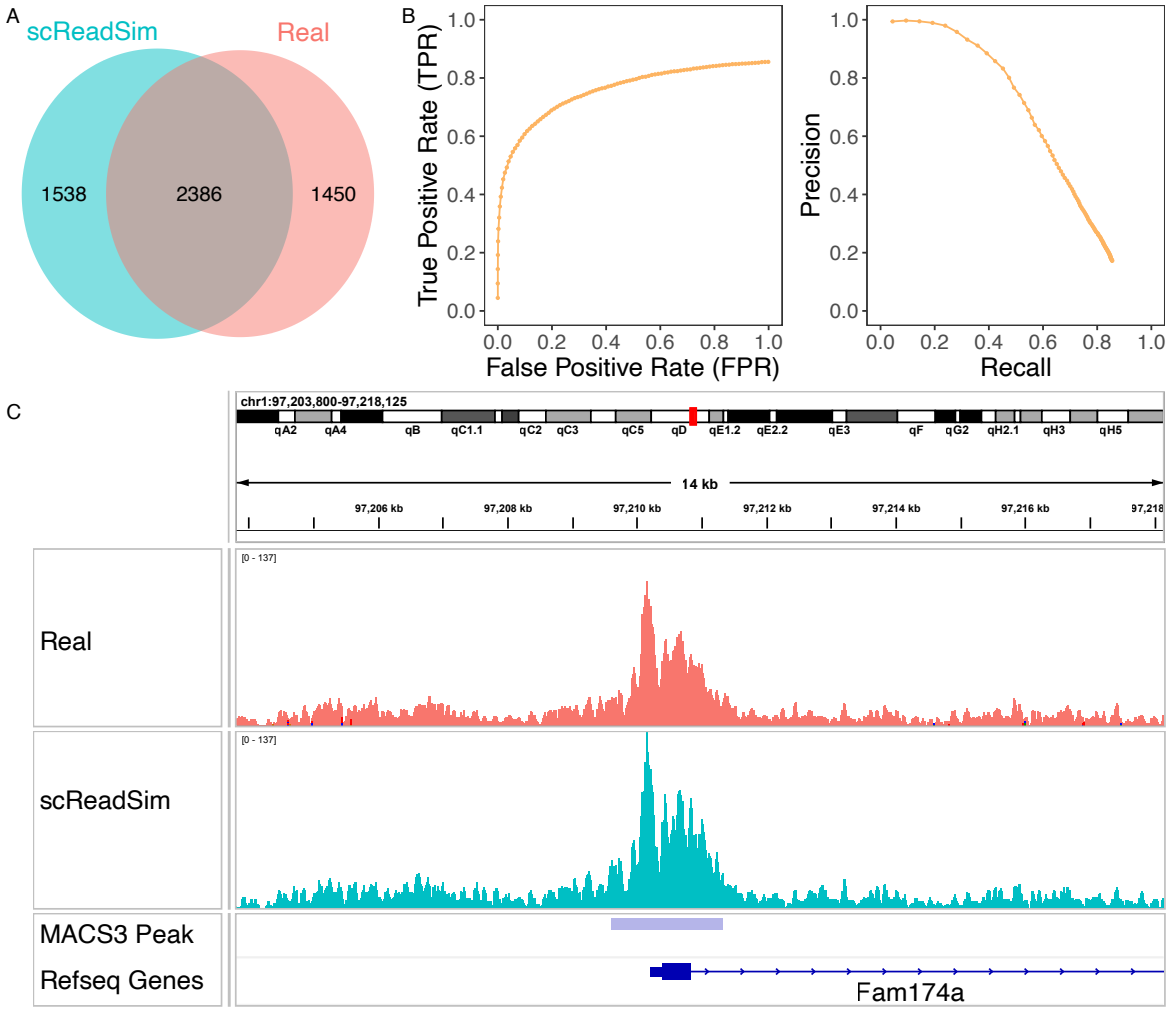

Figure S2: Synthetic reads generated by scReadSim resemble real sci-ATAC-seq data in terms of chromatin open regions. **A** Venn diagram of peak calling results using MACS3 on both scReadSim synthetic and real BAM files. **B** True positive rate (TPR) vs. false positive rate (FPR) and precision vs. recall by taking real peaks as truths. The peak identified using a synthetic BAM file is taken as a true peak if the overlapped length between it and one real peak is larger than half of the minimal scReadSim peak length. **C** Read coverage comparison through IGV genome browser. BAM files of the scReadSim synthetic data and real data are aligned side by side onto the reference genome. The peak regions called using MACS3 on real BAM file are shown in the MACS3 Peaks track.

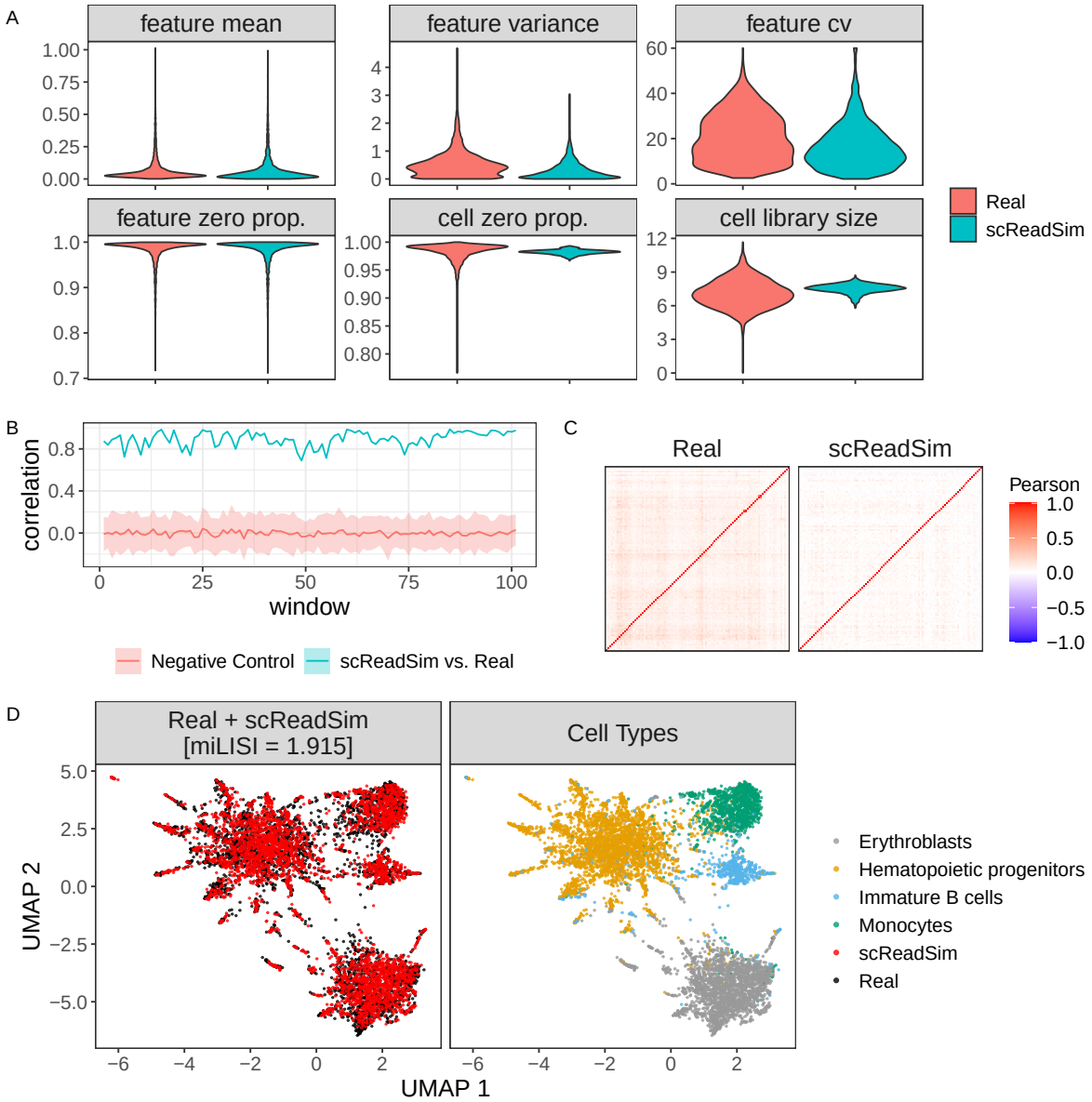

Figure S3: Synthetic reads generated by scReadSim resemble real sci-ATAC-seq data in count level. **A** Summary statistics of the count matrix for feature level (mean, variance, coefficient of variance (cv), zero proportion) and cell level (zero proportion and cell library size). **B** Correlation of read coverage between scReadSim synthetic and real BAM among 100 windows along the reference genome. The negative control group in red color represents the correlation of real BAM read coverage between random pairs of windows. Random sampling repeats 100 times for each window, and the mean and standard deviation are reported. **C**. Correlation among 100 top-expressed features using the synthetic and real count matrix. The top-expressed features are selected using the real count matrix. **D** UMAP visualization of the mixture of real and scReadSim synthetic count matrices. Cells in the left figure are labeled by the dataset source: real or synthetic cells. In the title of the left figure, miLISI measures the mixture of synthetic and real data points, which is close to 2 if they are perfectly mixed. Cells in the right figure are labeled by the cell types predefined in the real dataset.

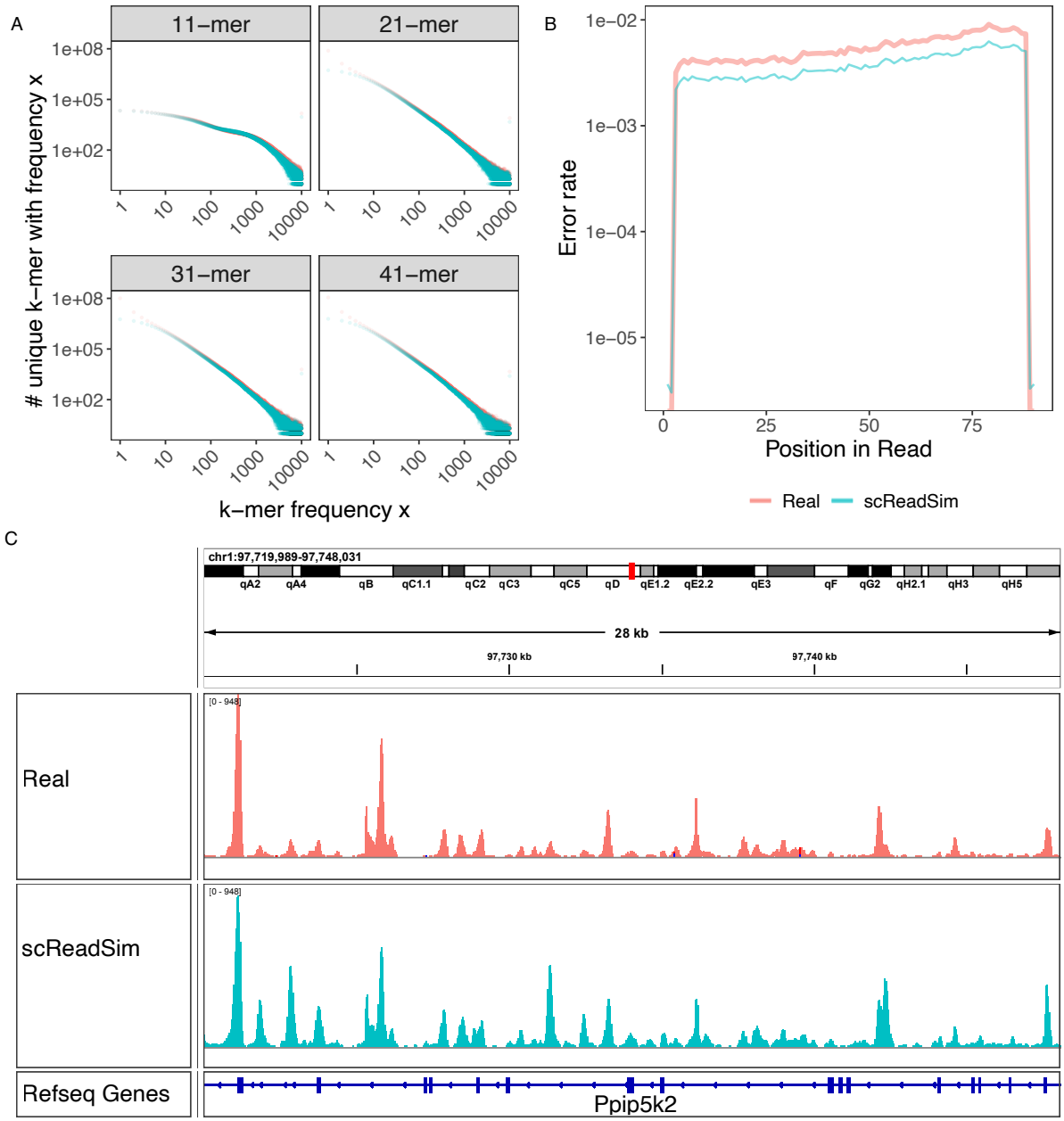

Figure S4: Synthetic reads generated by scReadSim resemble real 10x scRNA-seq data in read level. **A** K-mer spectrum comparison between scReadSim synthetic and real data. The x-axis refers to the frequency that occurred for a specific k-mer, while the y-axis represents the number of unique k-mers with this frequency. Both x-axis and y-axis are in log_{10} scale. **B** Error rate per base call within the read. The line plot's x-axis represents the positions of bases within a read, and the y-axis refers to the substitution error rates corresponding to that position. **C** Read coverage comparison through IGV genome browser. BAM files of the scReadSim synthetic data and real data are aligned side by side onto the reference genome.

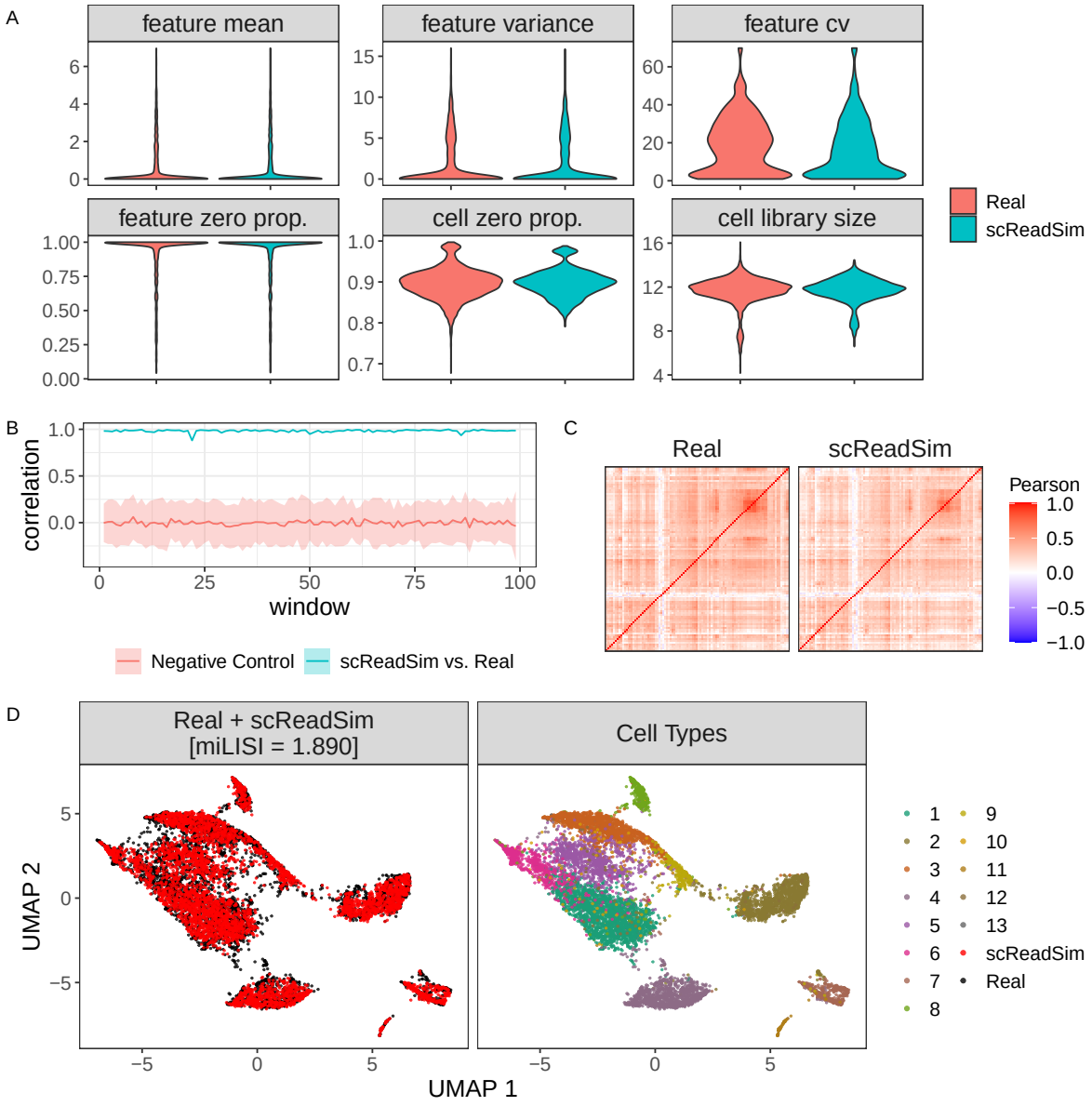

Figure S5: Synthetic reads generated by scReadSim resemble real 10x scRNA-seq data in count level. **A** Summary statistics of the count matrix for feature level (mean, variance, coefficient of variance (cv), zero proportion) and cell level (zero proportion and cell library size). **B** Correlation of read coverage between scReadSim synthetic and real BAM among 100 windows along the reference genome. The negative control group in red color represents the correlation of real BAM read coverage between random pairs of windows. Random sampling repeats 100 times for each window, and the mean and standard deviation are reported. **C**. Correlation among 100 top-expressed features using the synthetic and real count matrix. The top-expressed features are selected using the real count matrix. **D** UMAP visualization of the mixture of real and scReadSim synthetic count matrices. Cells in the left figure are labeled by the dataset source: real or synthetic cells. In the title of the left figure, miLISI measures the mixture of synthetic and real data points, which is close to 2 if they are perfectly mixed. Cells in the right figure are labeled by the cell clusters estimated during the synthetic count matrix generation process within scReadSim.

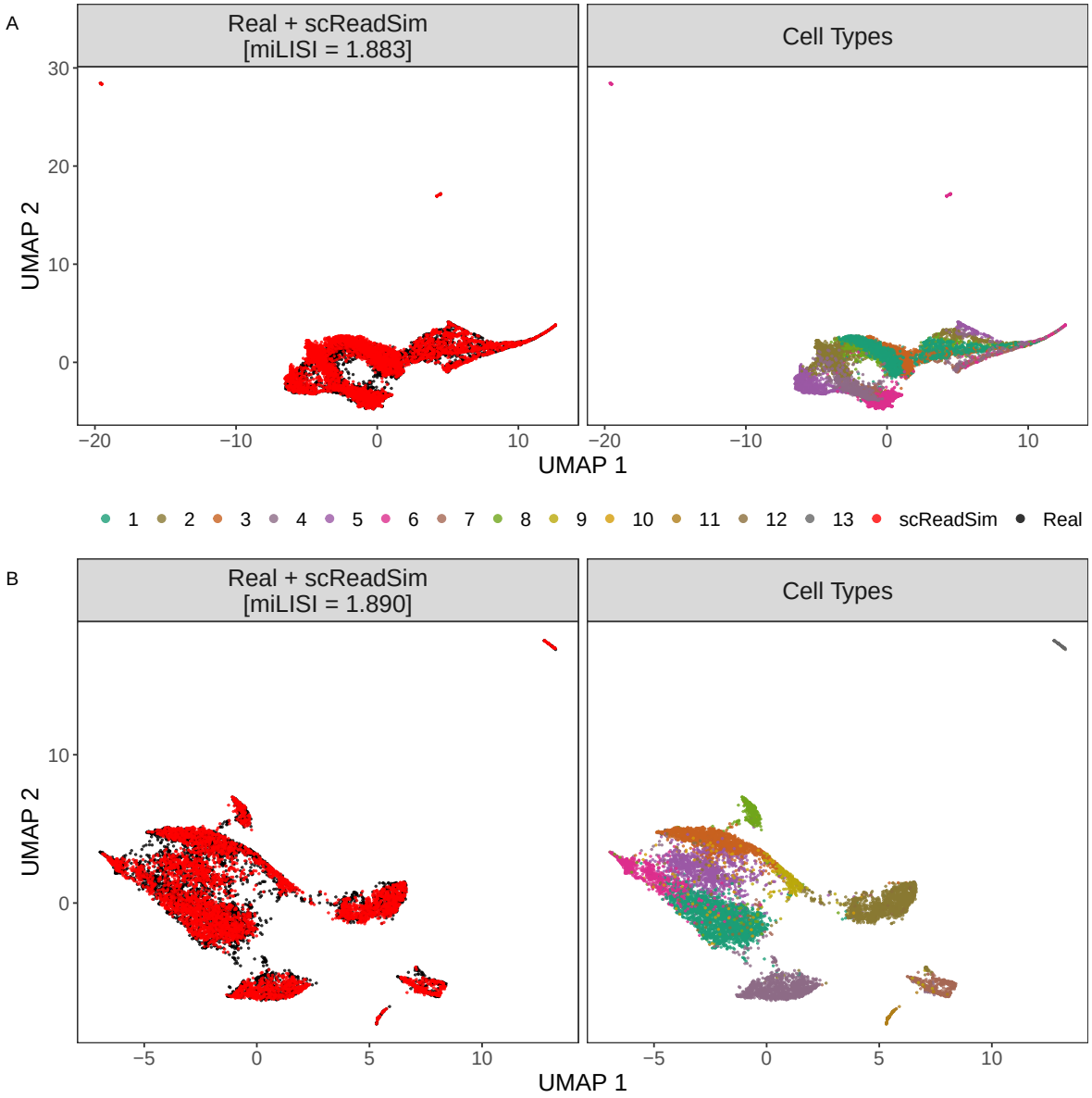

Figure S6: UMAP visualization of synthetic 10x scATAC-seq and scRNA-seq data with outliers. **A** UMAP visualization of the mixture of real and scReadSim synthetic count matrices. This is a zoom-out version of Fig. 5D (Left) without removing any outliers. **B** UMAP visualization of the mixture of real and scReadSim synthetic count matrices. This is a zoom-out version of Fig. S5D (Right) without removing any outliers.

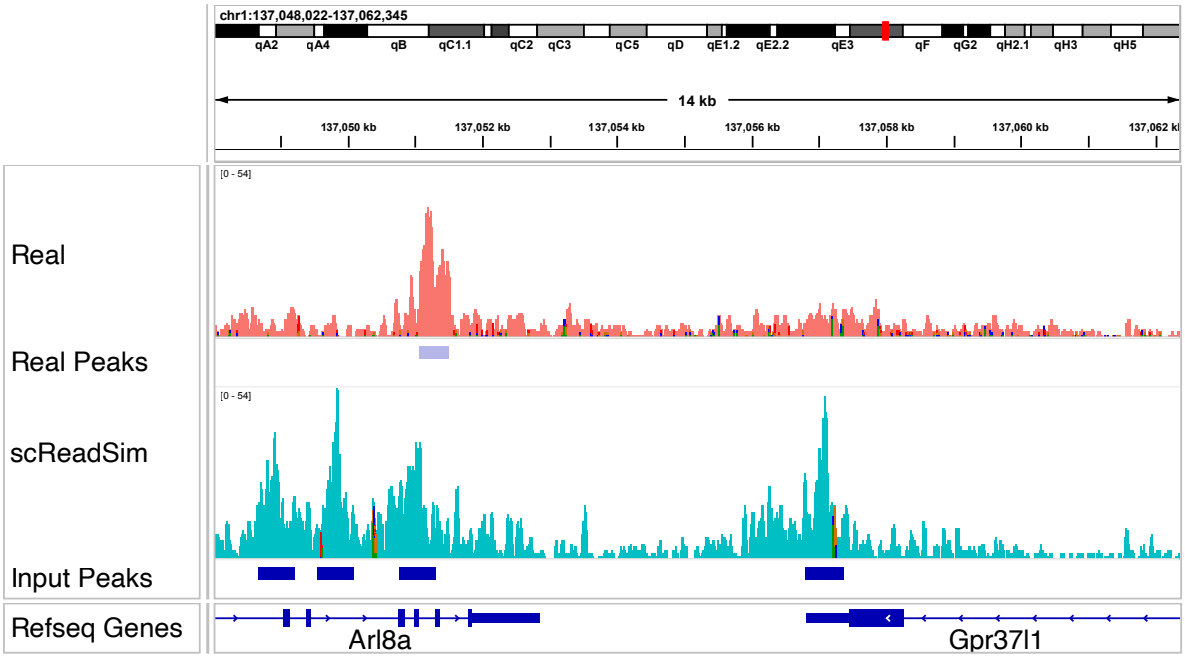

Figure S7: Visualization of Synthetic reads generated by scReadSim with user-specified chromatin open regions in IGV genome browser. Read coverage comparison through IGV genome browser. BAM files of the scReadSim synthetic data and real data are aligned side by side onto the reference genome. The peak regions called using MACS3 on the real BAM file are shown in the Real Peaks track underneath the real BAM read coverage track. In contrast, the peaks specified by truncating genes' transcript regions (input peaks) are shown in the User Input Peaks track underneath the scReadSim BAM read coverage track.

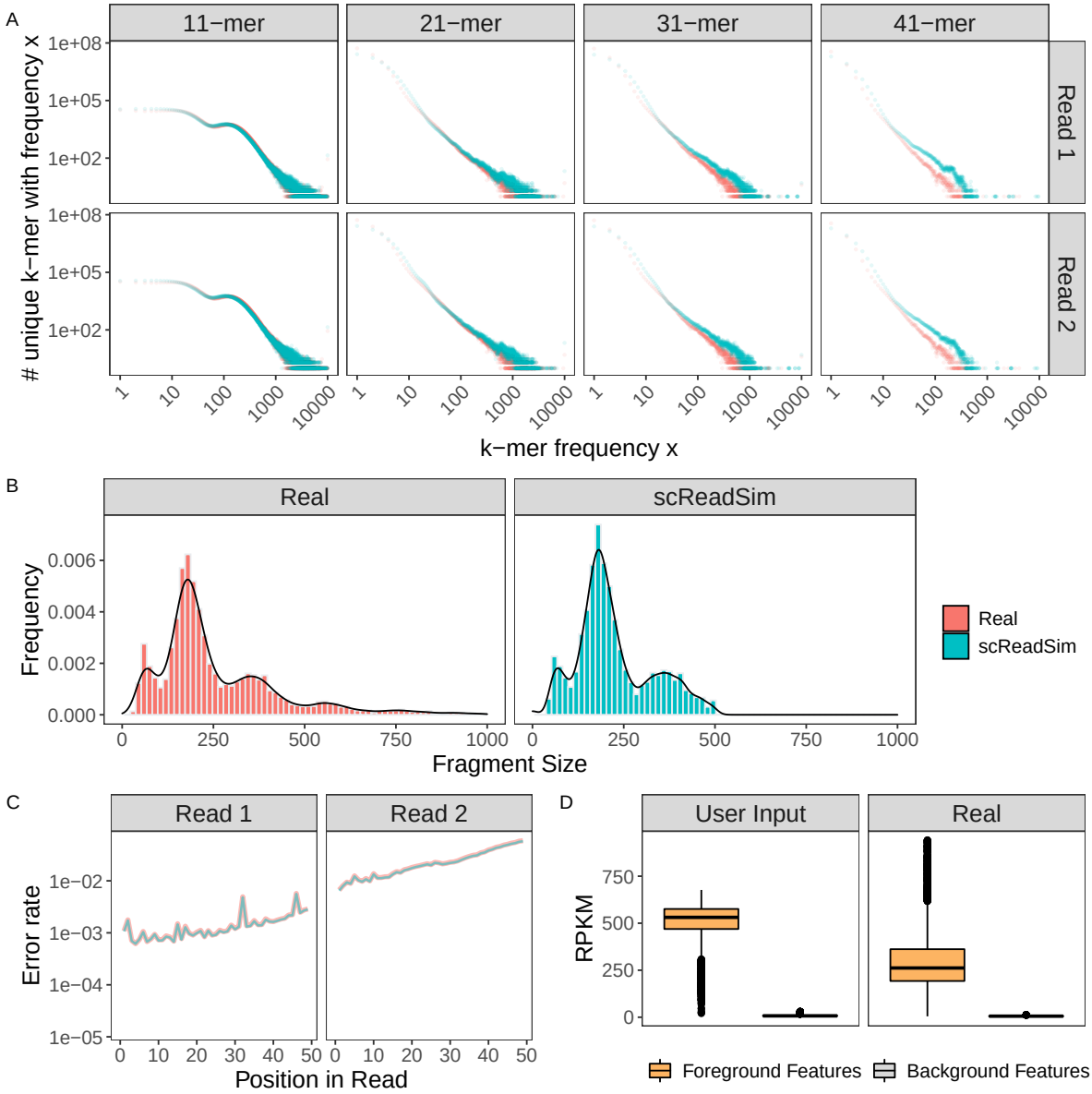

Figure S8: Synthetic reads generated by scReadSim with user-specified chromatin open regions resemble real sci-ATAC-seq data in read level. **A** K-mer spectrum comparison between scReadSim synthetic and real data. The x-axis refers to the frequency that occurred for a specific k-mer, while the y-axis represents the number of unique k-mers with this frequency. Both the x-axis and y-axis are in $\log_{10}$ scale. **B** Fragment length distribution comparison between scReadSim synthetic and real data. **C** Error rate per base call within the read. The line plot's x-axis represents the positions of bases within a read, and the y-axis refers to the substitution error rates corresponding to that position. **D** Reads Per Kilobase Million (RPKM) of scReadSim synthetic and real BAM files. Features are categorized into the foreground and background features. The left figure is for synthetic data with foreground features denoting the user-input peaks. The right figure is for real data with foreground features as peaks identified using MACS3 on the real BAM file. The background features are the complementary regions of foreground features.

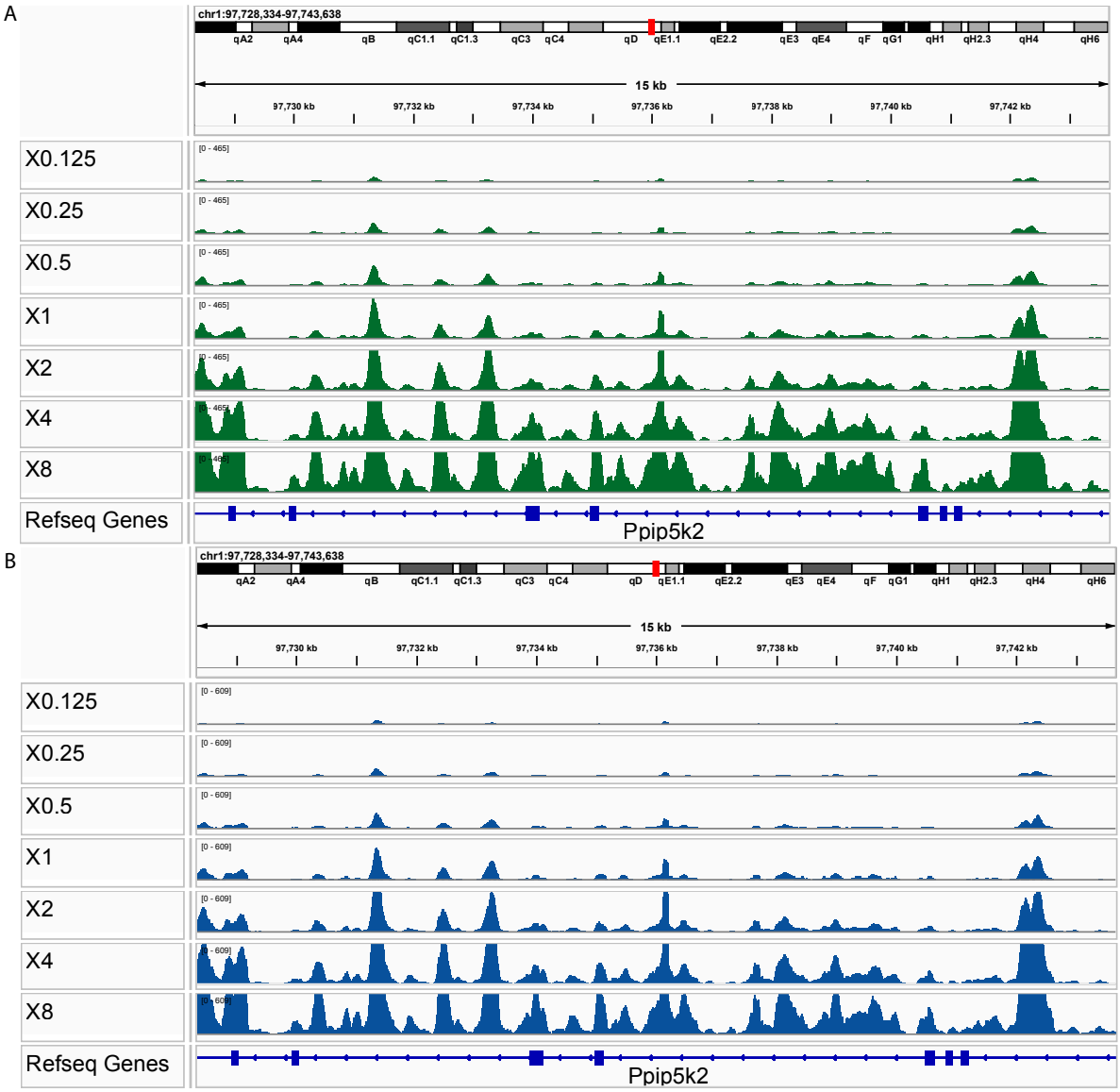

Figure S9: scReadSim is flexible for allowing varying throughputs and library sizes as input parameters to guide experimental design. Synthetic BAM files generated by scReadSim from 10x scRNA-seq data shown in IGV genome browser. **A** Synthetic BAM files generated by scReadSim with different amounts of synthetic cells. The names of tracks indicate the multiplication factors when specifying the synthetic cell number. Note that $\times 1$ means using the original cell number, while $\times 8$ means using 8 times as many as the original cell number. **B** Synthetic BAM files generated by scReadSim with different amounts of sequencing depths. The names of tracks indicate the multiplication factors when specifying the synthetic sequencing depth. Note that $\times 1$ means using the original sequencing depth, while $\times 8$ means using 8 times as many as the original sequencing depth.