scMultiSim: simulation of single cell multi-omics and spatial data guided by gene regulatory networks and cell-cell interactions

Xiuwei Zhang (xiuwei.zhang@gatech.edu)
Georgia Institute of Technology https://orcid.org/0000-0002-1713-772X

Hechen Li
City University of Hong Kong

Ziqi Zhang
Georgia Institute of Technology https://orcid.org/0000-0002-8198-0260

Michael Squires
Georgia Institute of Technology

Xi Chen
Southern University of Science and Technology https://orcid.org/0000-0003-2648-3146

Article

Keywords:

Posted Date: September 19th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3301625/v1

Additional Declarations: There is **NO** Competing Interest.

scMultiSim: simulation of single cell multi-omics and spatial data guided by gene regulatory networks and cell-cell interactions

Hechen Li[1], Ziqi Zhang[1], Michael Squires[1], Xi Chen[2], and Xiuwei Zhang[1,*]

[1]Georgia Institute of Technology, Atlanta, USA
[2]Southern University of Science and Technology, Shenzhen, China
[*]Corresponding author: xiuwei.zhang@gatech.edu

Simulated single-cell data is essential for designing and evaluating computational methods in the absence of experimental ground truth. Existing simulators typically focus on modeling one or two specific biological factors or mechanisms that affect the output data, which limits their capacity to simulate the complexity and multi-modality in real data. Here, we present scMultiSim, an *in silico* simulator that generates multi-modal single-cell data, including gene expression, chromatin accessibility, RNA velocity, and spatial cell locations while accounting for the relationships between modalities. scMultiSim jointly models various biological factors that affect the output data, including cell identity, within-cell gene regulatory networks (GRNs), cell-cell interactions (CCIs), and chromatin accessibility, while also incorporating technical noises. Moreover, it allows users to adjust each factor's effect easily. We validated scMultiSim's simulated biological effects and demonstrated its applications by benchmarking a wide range of computational tasks, including multi-modal and multi-batch data integration, RNA velocity estimation, GRN inference and CCI inference using spatially resolved gene expression data, many of them were not benchmarked before due to the lack of proper tools. Compared to existing simulators, scMultiSim can benchmark a much broader range of existing computational problems and even new potential tasks.

Hechen Li *et al.* | August 28, 2023 | 1–59

Introduction

In recent years, technologies that profile the transcriptome and other modalities (multi-omics) of single cell have brought remarkable advances in our understanding of cellular mechanisms [65]. For example, technologies have enabled the joint profiling of chromatin accessibility and gene expression data [13; 11; 46], and spatial locations of cells can be measured together with transcriptome profiles using imaging-based [57; 23; 67] or sequencing-based [60; 55] technologies.

The advent of single-cell multi-omics data has facilitated a more comprehensive understanding of cellular states, and more importantly, allowed researchers to explore the relationships between modalities and the causality across hierarchies [22]. Prior to the availability of single cell multi-omics data, gene regulatory network (GRN) inference methods were developed using only single-cell RNA sequencing (scRNA-seq) data [53]. However, these methods mainly focused on transcription factors (TFs) as the sole factor affecting gene expressions. In reality, the observed gene-expression data is affected by multiple factors, such as the chromatin accessibility of corresponding regions. Consequently, newer methods utilizing both scRNA-seq and scATAC-seq data have been developed to infer GRNs [4; 34; 66; 73]. Similarly, there has been a surge in the development of other computational tools that harness multi-modality information. For instance, Cell-Cell Interaction (CCI) inference methods seek to utilize both the gene expression and the spatial location modalities [20; 58; 8**?**] to learn the interactions with a lower false-positive rate than those using only scRNA-seq data [7; 30; 33]. Data integration methods combine multi-omics data to obtain a wholistic view of cells [62; 68; 2; 75; 40]. Moreover, RNA velocity can be inferred from unspliced and spliced counts using scRNA-seq data to indicate the near-future state of each cell [39; 6]. Recently, methods have also been proposed to infer RNA velocity from jointly profiled chromatin accessibility and transcriptomics data [42].

Overall, a large number of computational methods have been developed using scRNA-seq data or single cell multi- and spatial-omics data [71]. However, the scarcity of *ground truth* in experimental data makes it difficult to evaluate the performance of proposed computational methods. To address this, *de novo* simulators have been widely used to evaluate the accuracy of computational methods by generating data that models biological mechanisms and provides ground truth for benchmarking. SymSim [74], for example, provides ground truth cell identity and gene identity and thus can benchmark clustering, trajectory inference and differential expression detection. SERGIO [18], BEELINE [53] and dyngen [10] can simulate scRNA-seq data with given ground truth GRNs for testing GRN inference methods; while SERGIO, dyngen and VeloSim [76] can provide ground truth RNA velocity for testing RNA velocity inference methods. mistyR [64] generates single cell gene expression data from a given CCI network and can test methods that infer CCIs between cell types. With the *de novo* simulators, users can easily control the input parameters and obtain the exact ground truth. In addition to *de novo* simulators, Crowell *et al* [15] discussed another category of single cell data simulators, namely the reference-based methods, which learn a generative model from a given real dataset and generate synthetic data [16; 63; 59; 5]. By design, these methods can output datasets that mimic the input reference data, but their flexibility can be limited by the availability of reference datasets, and extracting ground truth information like GRNs, CCIs or RNA velocity remains a challenge for these simulators.

We consider that a desirable single cell simulator should meet several criteria: (1) it should generate as many modalities as possible to best represent a cell; (2) it should model as many biological factors and mechanisms that affect the output data as possible so that the output data has realistic complexity; and (3) it should provide ground truth of the biological factors to benchmark various computational methods. Most existing simulators generate only scRNA-seq data, and some generate only scATAC-seq data [50; 41]. Among the few ones that can generate multiple modalities, dyngen and SERGIO output unspliced and spliced counts with ground truth RNA velocity, while a reference-based simulator scDesign3 [59] can generate two modalities each with high dimensionality (*eg.* scRNA-seq and DNA methylation data), or one high-dimensional modality (*eg.* scRNA-seq) and spatial location data depending on the input reference dataset (Table S1).

In terms of the biological factors modeled in the simulator, existing *de novo* simulators model only one or a small subset of the following biological factors that affect gene expression in a cell: cell identity (cluster labels or positions on cell trajectories), chromatin accessibility, GRNs, and CCIs (Table S1). Data generated by reference-based simulators can inherently have these effects but it is challenging to obtain the ground truth of the biological factors, thus unable to measure the accuracy of a computational method.

In this paper, we present scMultiSim, a unified framework that models *all* the above biological factors as well as technical variations including sequencing noise and batch effect. For each single cell, it outputs the following modalities: unspliced and spliced mRNA counts, chromatin accessibility, and spatial location, while considering the cross-modality relationships (Fig. 1a). "Chromatin accessibility" is both an output modality (also called the scATAC-seq modality) and a biological factor that affects other output data (it affects the gene expression modality).

scMultiSim provides ground truth information on cell identity (in terms of cell populations), RNA velocity, GRNs and CCIs, as well as relationships between chromatin accessibility and transcriptome data. Therefore, with one dataset, it can be used to evaluate methods for various computational tasks, including clustering or trajectory inference, multi-modal and multi-batch data integration (mosaic integration), RNA velocity estimation, GRN inference and CCI inference (Fig. 1a). We show that scMultiSim can be used to test methods that can not be benchmarked with existing simulators, including methods for mosaic integration, GRN inference using multi-omics data, and CCI inference between single cells; For computational tasks with existing benchmarking efforts (GRN inference using only scRNA-seq data, clustering, trajectory inference, RNA velocity), we show that we obtain overall consistent results or new insights. Moreover, scMultiSim allows the users to adjust the effect of each biological factor on the output data, enabling them to investigate how the methods' performance is affected by each factor when evaluating methods for a specific task. To our knowledge, scMultiSim is the most versatile simulator to date in terms of its benchmarking applications (Table S1).

Results

In the following sections, we will provide a brief overview of the core concepts and the simulation process of scMultiSim. We will then demonstrate its capability to simulate multiple biological factors simultaneously by validating the effects of each factor on the output data. Furthermore, we will showcase the applications of scMultiSim by using it to benchmark a wide variety of computational tools.

scMultiSim overview

The kinetic model and control of intrinsic noise. In general, scMultiSim runs the simulation in two phases (Fig. 1b). In the first phase, scMultiSim employs the widely-accepted kinetic model [52] to generate the true gene expression levels in cells ("true counts"). In the second phase, scMultiSim introduces technical variations (library preparation noise, batch effects, etc) and generate scRNA-seq and scATAC-seq data that are statistically comparable to real data ("observed counts"). To model cellular heterogeneity and gene regulation effects, scMultiSim introduces two main concepts: *Cell Identity Factors* (CIFs) and *Gene Identity Vectors* (GIVs) (Fig. 1b (i, ii)). Biological factors, including cell population (cell identity), GRNs, and CCIs, are encoded in CIFs and GIVs (Fig. 2a). Additionally, to model single-cell chromatin accessibility, we also introduce Region Identity Vectors (RIVs, Fig. 1b(iii)). Further details on CIF, GIV and RIVs are provided in the next section.

When simulating single cell gene expression data, scMultiSim extends the idea of SymSim [74], where a kinetic model with three major parameters k_{on}, k_{off}, s was used to determine the expression pattern of a gene in a cell (Fig. 1b (vi)). In the kinetic model, a gene can switch between *on* and *off* states, with k_{on} and k_{off} be the rates of becoming *on* and *off*. When a gene is in the *on* state (which can be interpreted as promoter activation), mRNAs are synthesized at a rate s and degrade at a rate d. It is common to fix d at 1 and use the relative values for the other three parameters [49]. The kinetic parameters k_{on}, k_{off}, s are calculated from the CIF and GIV, as well as the corresponding scATAC-seq data (because chromatin accessibility is considered to affect gene expression). Since GIVs and CIFs encode information on cell identity, GRNs, and CCIs, the kinetic parameters thus capture the four biological factors that affect gene expression: cell identity, chromatin accessibility, GRNs, and CCIs.

The kinetic model used in scMultiSim provides two modes for generating true counts from the parameters, as shown in Fig. 1b (vii). The first mode is the *full kinetic model*, where genes undergo several cycles of *on*/*off* state changes over time, and the spliced/unspliced RNA counts are generated. This mode provides ground truth RNA velocity. The second mode is the *Beta-Poisson model*, which is equivalent to the kinetic model's master equation [35], and is faster to run than the full kinetic model. The Beta-Poisson model is recommended when RNA velocity is not needed. scMultiSim also introduces an intrinsic noise parameter σ_i that controls the amount of intrinsic noise caused by the transcriptional burst and the snapshot nature of scRNA-seq data. This parameter allows users to examine the influence of intrinsic noise on the performance of the computational methods. The two modes and the σ_i parameter are further described in Methods.

Modeling cellular heterogeneity and various biological effects. The design of *Cell Identity Factors (CIFs)* and *Gene Identity Vectors (GIVs)* allows scMultiSim to encode cell identities and gene-level mechanisms (such as GRNs and CCIs) into the kinetic parameters and thereby impact the gene expression levels. This design also provides easy ways to adjust the effect of each factor on the output gene expression data.

The CIF of a cell is a 1D vector representing various biological factors that contributes to cellular heterogeneity, such as the cell condition (*e.g.* treated or untreated), or the expression of key TFs. The GIV of a gene act as the weights of the corresponding factors in the CIF, representing how strongly the corresponding CIF affect the gene's expression (Fig. 2a, Methods). By multiplying the CIF and GIV matrices, scMultiSim therefore generates a $n_{\text{cell}} \times n_{\text{gene}}$ matrix, which is the desired kinetic parameter matrix with the cell and gene factors encoded.

Each CIF vector and GIV vector consists of four segments, each representing one type of extrinsic variation. They encode biological factors including cell identity (cell population, *i.e.*, the underlying cell trajectories or clusters), GRNs, and CCIs (Figs. 2a, S1a-b). We introduce the four segments in the following.

(i) Non-differential CIFs (**non-diff-CIF**) model the inherent cellular heterogeneity. They represent various environmental factors or conditions that are shared across all cells and are sampled from a Gaussian distribution with standard deviation σ_{cif}.

(ii) Differential CIFs (**diff-CIF**) control the user-desired cell populations. When generating data for cells from more than one cell type, the minimal user input of scMultiSim is the cell differentiation tree (default trees are provided), which controls the cell types (for discrete populations) or trajectories (for continuous populations) in the output. The cell differentiation tree is used to generate diff-CIFs (Methods).

(iii) CIFs corresponding to Transcription Factors (**tf-CIF**) control the effects of GRNs. This segment, together with the TF segment in the GIV, model how a TF can affect expression of genes in the cell (Methods). Its length equals to the number of TFs. In other words, the GRN is encoded in the tf-CIFs and GIVs.

(iv) CIFs corresponding to ligands from neighboring cells (**lig-CIF**) control the effect of CCI. If CCI simulation is enabled, this segment together with the ligand segment in the GIV of the receptor gene encodes the ground truth CCI between two cells (Methods). This encoding ensures that a ligand and its interacting receptor have correlated gene expression. A cell can interact with multiple other cells (Fig. 2a (viii)).

Simulating spatial data. If specified to generate spatial-aware single cell gene expression data including cell spatial locations and CCI effects, scMultiSim uses a multiple-step approach that considers both time and space (Fig. 1b (viii), Fig. S1c). The simulation consists of a series of steps, with each step representing a time point. Cells are placed in a grid (Fig. 2a (ix), Fig. S1d), and one cell is added to the grid at each step, representing a newborn cell. Users can use the parameter p_n to control the probability for the newborn cell to locate with cells of the same type (Methods). As experimental data cannot measure cells at previous time points, scMultiSim outputs data only for cells at the final time point, which contains the accumulated CCI effects during the cells' developmental process.

The output spatial data can have different layout of cell types mimicking different tissues, and users can choose from "default", "layers" and "islands" (Methods).

To simulate CCI, scMultiSim requires a user-inputted list of ligand-receptor gene pairs that can potentially interact, which is called a ligand-receptor database. Users can input cell-type-level or single cell level CCI ground truth. If users do not provide ground truth CCIs, scMultiSim can randomly generate the ground truth from the ligand-receptor database. scMultiSim allows users to simulate both the long-range and short-range CCIs as described in [44] (Methods).

Technical variations and batch effects. The steps described above belong to the first phase, which generates the "true" mRNA counts (and unspliced counts if RNA velocity mode is enabled) in the cells. In the second phase, scMultiSim simulates key experimental steps in wet labs that lead to technical noises in the data and output the observed scRNA-seq data (Fig. 1b, Phase 2). Batch effects can be added to simulate datasets from a user-specified number of batches. Users can also control the amount of technical noise and batch effects between batches. These procedures are described in Methods.

The overall simulation process. Fig. 1b shows an overview of the simulation process. The scATAC-seq data is generated at first (Fig. 1b(iv)), because we consider that the chromatin accessibility of a cell affects its gene expression. The scATAC-seq data also follows a pre-defined clustering or trajectory structure represented by the input cell differentiation tree. Details on generating the scATAC-seq data are included in Methods. The scATAC-seq data affects scRNA-seq data through the k_{on} parameter, because chromatin accessibility controls the activated status of genes (Methods). A TF-motif matrix and the ground truth GRN are considered when determining the chromatin regions that control a given gene (Methods). After obtaining all the kinetic parameters, scRNA-seq data can be generated in different modes: with or without CCIs and spatial locations, and with or without outputting RNA velocity data (Fig. 1b (vii, viii)). Finally, technical noise and batch effects are added to the "true counts" generated from Phase 1. Next, we show the various output of scMultiSim and validate the effects present in the simulated data.

Design of simulation and datasets

We have generated a comprehensive set of datasets using scMultiSim to demonstrate the effects of different parameter configurations and to benchmark computational methods. These datasets contain both *main* and *auxiliary* datasets. The *main* datasets contain all effects scMultiSim can simulate: GRN, chromatin accessibility, cell-cell interaction, technical noise and batch effect. These main datasets consist of 144 datasets with varying configurations of important parameters, including σ_{cif} (which controls the standard deviation of the CIF and affects the within-cluster or within-neighborhood heterogeneity between cells), the numbers of cells (n_{cell}) and genes (n_{gene}), and three different cell population structures (Table 1). Thus, the 144 main datasets cover a wide range of variety, including different numbers of cells, genes, and trajectory shapes, to minimize potential bias and provide a more comprehensive benchmark of the computational methods.

As presented in Table 1, we label the main datasets with the following format: `M{p}{c}{s}`. The first letter M denotes the main dataset, followed by a letter `p` $\in \{L, T, D\}$ that specifies the cell population as linear trajectory, tree trajectory or discrete, respectively. The number `c` $\in [1, 12]$ denotes a particular configuration of σ_{cif}, n_{cell}, and n_{gene}, while the last lowercase letter `s` $\in \{a, b, c, d\}$ represents random seed 1-4. For instance, the dataset `MD5c` has a discrete cell population, $\sigma_{\text{cif}} = 0.1$, 800 cells, 200 genes and random seed 3.

We have also generated multiple *auxiliary* datasets with fewer types of effects and presented them in Table 2. These datasets allow us to explore the effect of certain attributes in the data (controlled by interpretable parameters) that can affect the performance of specific computational methods, and it is faster to generate large auxiliary datasets than main datasets. In the remaining, we will primarily use the main datasets `M` for benchmarking and demonstration, while the auxiliary datasets will provide additional and supplementary results.

scMultiSim generates multi-batch and multi-modality data from pre-defined clusters or trajectories

scMultiSim offers a key advantage in its ability to generate coupled scRNA-seq and scATAC-seq data while allowing users to control the shape of trajectories or clusters. First, the user can choose to generate “continuous” or “discrete” populations, and input a differentiation tree that represents the cell trajectories (in the case of “continuous” populations) or relationship between clusters (in the case of “discrete” populations). scMultiSim provides three example differentiation trees: Phyla1, Phyla3, and Phyla5, each having 1, 3, and 5 leaves, as illustrated in Fig. 2b. The main datasets were simulated using these trees (Table 1). From a differentiation tree, scMultiSim is able to generate both discrete and continuous cell populations (Fig. 2c-d). Then, users can use these three parameters: intrinsic noise σ_i, CIF sigma σ_{cif} and Diff-to-nonDiff CIF ratio r_d, to control how clean or noisy the population structure is in the data (Fig. 2c-e).

For the continuous population, we visualize a dataset `MT3a` generated using tree Phyla3 in Fig. 2c. We can observe that the trajectories corresponding to the input differentiation tree are clearly visible for both the scRNA-seq and the scATAC-seq modality. For the discrete population, we visualize dataset `MD3a` and `MD9a` generated with tree Phyla5 in Fig. 2d. The parameter σ_{cif} controls the standard deviation of the CIF, therefore with a smaller σ_{cif}, the clusters are tighter and better separated from each other. We then used the auxiliary dataset `A` (Table 2) to explore the effect of the intrinsic noise parameter σ_i and r_d, the ratio of number of diff-CIF to non-diff-CIFs. In Fig. 2e, we visualize the scRNA-seq modality generated using Phyla5 continuous mode with the same σ_{cif}. With a smaller Diff-to-nonDiff CIF ratio r_d, the trajectory is vague and more randomness is introduced, as the tree structure is encoded in the differential CIFs. With a smaller intrinsic noise σ_i the trajectory is more prominent. These patterns are cleaner than real data because real data always has technical noise. We will show more results with technical noise in later sections and in Fig. S2.

Coupling between scATAC-seq and scRNA-seq data. In paired scATAC-seq and scRNA-seq data, these two data modalities are not independent of each other, as it is commonly considered that a gene's expression level is affected by the chromatin accessibility of the corresponding regions. If a gene's associated regions are accessible, this gene is more likely to be expressed. This mechanism can be naturally modeled in scMultiSim through the kinetic parameter k_{on} (Methods).

We provide a user-adjustable parameter, the ATAC-effect E_a, to control the extent of scATAC-seq data's effect on k_{on} (ranging from 0 and 1). In order to validate the connection between the scATAC-seq and scRNA-seq data, we calculate the mean Spearman correlation between these two modalities for genes that are controlled by one region in the scATAC-seq data. In Fig. 2f, we present the correlations under different E_a values. An averaged 0.2-0.3 correlation can be observed using the default value (0.5), and the correlation increases with higher values of E_a. These results demonstrate that scMultiSim successfully models the connection between scATAC-seq and scRNA-seq data, enabling the generation of more realistic multi-omics datasets.

scMultiSim simulates technical noise and batch effect. The single cell gene expression data shown in Figs. 2c-f are "true" mRNA counts which do not have technical noise. scMultiSim can add technical noise including batch effects to the true counts to obtain observed counts (Methods). The amount of technical noise and batch effects can be adjusted through parameters, for example, the parameter E_{batch} can be used to control the amount of batch effect. Users can also specify the number of batches.

Fig. 2g shows the observed mRNA counts of dataset `MD9a` (true counts shown in Fig. 2d). The left plot shows data with one batch, and the right plot shows two batches. Technical noise and batch effects are also added to the scATAC-seq matrix. We further use the auxiliary dataset `A` to demonstrate the ability of scMultiSim to adjust the amount of technical noise and batch effect in both scRNA-seq and scATAC-seq modalities, in both continuous and discrete populations (Fig. S2). Here, we vary a major parameter for technical noise, α, which denotes the mRNA capture efficiency where lower α corresponds to poorer detection ability of the dataset.

scMultiSim generates spliced and unspliced mRNA counts with ground truth RNA velocity

If RNA velocity simulation is enabled, the kinetic model outputs the velocity ground truth using the RNA splicing and degradation rates. The Phyla5 tree in Fig. 2b is used to generate the results in Fig. 2h. The figure shows both the true spliced and unspliced counts, as well as the ground truth RNA velocity averaged by k nearest neighbor (kNN), which can be used to benchmark RNA velocity estimation methods. The RNA velocity vectors follow the cell trajectory (backbone and directions shown in red) specified by the input differentiation tree.

scMultiSim generates single cell gene expression data driven by GRNs and cell-cell interactions

The strength of scMultiSim also resides in its ability to incorporate the effect of GRN and CCI while preserving the pre-defined trajectory structures. In this section, we show that the GRN and CCI effects both exist in the simulated expression data. The main datasets (Table 1) used the 100-gene GRN from [18] as the ground truth GRN, which is visualized in Fig. 3a. We also incorporate CCIs by adding cross-cell ligand-receptor pairs to the within-cell GRNs. Specifically, we connect each cell's gene 99,101-104 to a neighbor cell's gene 91, 2, 6, 10 (TFs), and 8 (non-TF) in the GRN (green edges in Fig. 3a). Next, we use one dataset (`MT3a` with a tree trajectory, 500 genes, 500 cells, and $\sigma_{\text{cif}} = 0.1$) to inspect the simulated effects in detail (Fig. 3b-e).

GRN guided expression data. We illustrate the gene regulation effects for dataset `MT3a` using a gene module correlation heatmap as shown in Fig. 3b. The clustered heatmap is constructed by computing pairwise Spearman correlations between the expression levels of all genes in the GRN. Each color on the top or left side of the heatmap represents a TF and its target genes in the GRN. The figure shows that gene modules regulated by the same TF (genes with the same color) tend to be clustered together and have higher correlations with each other. These results suggest the presence of GRN effects in the expression data. To further illustrate the regulatory effects, we plot the expression of a specific regulator-target pair (gene 19-20) along one lineage (4-5-3 in Phyla3) in Fig. 3c. The plot clearly shows a correlation between the expression levels of the regulator and target genes. Moreover, we include the accessibility levels for the corresponding chromatin region of gene 19 in this plot. The plot indicates that significant drops in gene 19's expression occur when the related chromatin region is closed, providing further evidence for the regulatory effects of chromatin accessibility.

Cell spatial locations. scMultiSim provides convenient helper methods to visualize the cell spatial locations, as shown in Fig. 3d (dataset `MT3a`). For each ligand-receptor pair, arrows can be displayed between cells to show the direction of cell-cell interactions. We consider various biological scenarios when assigning the spatial location to each cell (Methods), for example, a newborn cell has a probability p_n of staying with a cell of the same type. Changing p_n allows us to generate different tissue layouts. In real data, how likely cells from the same cell type locate together depends on the tissue type, and scMultiSim provides p_n to tune this pattern. Fig. 3f shows the effect of varying p_n. The left figure in the panel was generated with $p_n = 1$, showing strong spatial clustering of cells from the same cell type. The right figure in the panel was generated with $p_n = 0.8$, where cells from the same cell type are more spread out to enable more interactions across cell types. Apart from this "default" layout, we also provide other layouts commonly seen in spatial transcriptomics data, such as "islands" and "layers" (Fig. S3).

Correlations between interacting ligands and receptors. scMultiSim simulates CCIs between single cells as well as between cell types. Computational methods that infer CCIs use different scoring functions. Some assume interacting ligands and receptors should have correlated expression (*correlation effect*) while some assume they should have high expression (*high-expression effect*) [3]. Existing simulators for spatial data were able to implement the high-expression effect [77; 44], but did not incorporate the correlation effect, which is less straightforward to simulate.

We validate the simulated correlation effects of scMultiSim by comparing the correlations of expression levels between (i) neighboring cells with CCIs, (ii) neighboring cells without CCIs, and (iii) non-neighbor cells (Methods; "neighbor" here means cells within the neighborhood range). As shown in Fig. 3e (using dataset `MT3a`), cells with CCIs have an average pairwise correlation of 0.1, whereas cells without CCIs exhibit approximately zero correlation, which is expected. We noticed that neighboring cells without CCIs have a higher correlation compared to non-neighbor cells without CCIs, which may be attributed to the dynamic nature of cell differentiation, where cells are evolving into new cell types over time, and CCI effects involved in an earlier cell type may remain in the final step.

Out of existing efforts that simulate spatial data, SRTsim [77] and the simulation procedure in Liu *et al* [44] add the high-expression effect to the data by increasing the gene expression levels of ground truth interacting ligands–receptor pairs. The expression levels of downstream genes of receptors were not adjusted accordingly. mistyR [64] uses partial differential equations to generate the abundance of ligands, receptors and downstream genes, given CCIs between cell types. The way that CCI effects are simulated in scMultiSim is advantageous over mistyR by generating not only cell-type-level CCIs, but also single-cell-level CCIs, and modeling non-linear effects on ligands from upstream genes.

scMultiSim simulated datasets match real data

We show that scMultiSim's output single cell gene expression data can statistically resemble real data. We selected four single cell datasets, two of which are paired multi-modal datasets (10x Multinome PBMC, ISSAC-seq [70]), and the other two are spatially-resolved gene expression datasets (MERFISH [48], seqFISH+ [23; 20]). Simulated data were generated using scMultiSim to match these real datasets (Methods). We used dyngen [10] as a baseline simulator to compare with, as it is also a *de novo* multi-modality simulator that shares a few functionalities with scMultiSim (Table S1).

For single cell gene expression modality, we compare the simulated data with real data in terms of the following properties: library size, zero counts per cell, zero counts per gene, mean count per gene, variation per gene, and the ratio between zero count and mean count per gene (Fig. 3g, S4). Fig. 3g shows the result for the seqFISH+ dataset. It can be observed that the library size, zero counts per cell, zero counts per gene and mean counts per gene simulated by scMultiSim are closer to that of real data than the dyngen simulated data, and both scMultiSim and dyngen are able to simulate data with realistic variation per gene. There is also usually a negative correlation between zero counts and mean counts in real data, and scMultiSim is able to simulate this relationship, matching well with the reference data. Other three datasets has similar results (Fig. S4).

When the real dataset has the scATAC-seq modality (10x Multinome and ISSAAC-seq), we also show that the generated scATAC-seq data resembles the real data. scMultiSim also provides a parameter "atac.density" to set the kernel density of the generated ATAC-seq data, making it easier to match with a reference dataset. Three attributes are used to compare simulated scATAC-seq data with real data: library size, peak mean, and cell sparsity (Fig. S4). The similarity of these metrics indicates scMultiSim can also generate realistic ATAC-seq data. The parameters and scripts used to generate simulated data for these four real datasets are provided in the GitHub repository.

Benchmarking computational methods using scMultiSim

We next show that scMultiSim can be used to benchmark a board range of computational tasks in single cell genomics, including mosaic data integration, GRN inference with single-modality or multi-modality data, inference of CCIs between cell types and single cells using spatially resolved single cell gene expression data, clustering, trajectory inference and RNA velocity estimation. Using scMultiSim, we studied the performance of several methods on each task, and also investigated the effect of particular parameters for some of the benchmarks. As far as we know, scMultiSim is the only simulator that can benchmark all these tasks. It is noteworthy that our intention is not to perform a comprehensive benchmarking analysis, but rather to show evidence of scMultiSim's broad applications. We anticipate that these benchmarks can encourage forthcoming researchers to discover more use cases of scMultiSim.

Benchmarking mosaic data integration methods

A number of computational methods have been proposed to integrate single cell genomics data from multiple modalities and multiple batches [2] (mosaic integration). We benchmarked three recently proposed methods that can integrate data matrices from multiple batches and modalities: Seurat bridge integration (Seurat-bridge) [28], UINMF [38] and Cobolt [25]. We use all 144 main datasets to test their performance under various types of cell population. Each main dataset is divided into three batches (batch effect parameter $E_b = 3$), then the scRNA-seq data from batch 2 and scATAC-seq data from batch 3 are dropped intentionally to mimic a real scenario where some modalities are missing in certain batches (Fig. 4a). Fig. 4b shows the t-SNE visualization of one of the datasets `MT10a`. We use the following metrics to evaluate the performance of the integration methods: Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) as the metrics for cluster identity preservation, and Graph Connectivity and Average Silhouette Width (ASW) as metrics for batch mixing (Methods). These metrics were used in a recent paper on benchmarking single cell data integration methods [45].

The result is shown in Fig. 4c. Since Seurat-bridge does not output the latent embedding for the "bridge" dataset (batch 1 in Fig 4a), only the two matrices from batches 2 and 3 (colored in Fig. 4a) were used for evaluation. We observe that UINMF has the best performance in terms of all four measurements. Seurat-bridge and Cobolt have comparable ARI and NMI but Cobolt has better batch mixing scores. When comparing the ARI and NMI scores for $\sigma_{\text{cif}} = 0.1$ and $\sigma_{\text{cif}} = 0.5$, one can observe that these cell identity preservation scores are higher with smaller σ_{cif}. Comparing different cell population structures, we see that continuous populations ("Linear" and "Tree") have lower ARI and NMI scores than discrete populations, potentially because that metrics like ARI and NMI are better suited for discrete populations.

We then ran the integration methods on a large dataset with 3000 cells and visualized the integrated latent embedding in Fig. S5, which helped us to understand each method's behavior. We noticed that while Seurat-bridge has lower graph connectivity and ASW scores, different batches are located closely (but do not overlap) in the visualized latent space. That the reference and query data in the latent space do not overlap can cause the low batch mixing scores, but may not affect the ability of label transfer.

Benchmarking GRN inference methods using single- or multi-modality data

scMultiSim provides a systematic way to benchmark GRN inference methods that use (1) single-cell gene expression data (termed *expression-based GRN inference*) or (2) single-cell multi-omics data (termed *multi-omics GRN inference)*. We used scMultiSim to benchmark 11 *expression-based GRN inference* methods that were compared in a previous benchmarking paper [53], and 2 recently published *multi-omics GRN inference* methods, scMTNI [73] and CellOracle [34] (Methods). We measured the inference accuracy using AUROC (area under receiver operating characteristic curve) and AUPRC (area under precision-recall curve) ratio (Methods). These metrics were also used in previous benchmarking work [53].

We tested the methods on the 144 main datasets (results shown in Fig. 5a) as well as auxiliary datasets `G` (Table 2, results shown in Fig. 5b) with a linear trajectory and without CCI effect (Methods). With the auxiliary datasets `G`, we tested the methods using true counts and observed counts respectively, which allows us to investigate the robustness of methods against technical noise in the data (Fig. 5b). We first look at the performance of the 11 *expression-based GRN inference* methods. We observed that PIDC [12] has the best overall performance, especially on true counts, followed by GENIE3 [31] and GRNBOOST2 [47]. We then examined the effect of technical noise on the performance of GRN inference methods. With observed counts, both the AUPRC ratio and AUROC value suffer from a decline, showing that it is harder to infer the GRN from noisy data. Using observed counts, PIDC remains the top-performing method, while SINCERITIES [51], PPCOR [36] and SINGE [17] surpass GENIE3 and GRNBOOST2, indicating that SINCERITIES, PPCOR and SINGE can be relatively more tolerant to technical noise than GENIE3 and GRNBOOST2.

Notably, the ranking of the 11 *expression-based GRN inference* methods tested on true counts (Figs. 5a-b) is largely consistent with the ranking reported in previous benchmark result [53] using data without dropout and generated from four different biological curated models. This not only confirms the previous results but also suggests that scMultiSim can generate GRN-guided gene expression data with comparable benchmarking capability to other simulated or semi-simulated data. Meanwhile, scMultiSim brings new insights through simulating more biological effects in the data: comparing performance on the main datasets (Fig. 5a) and auxiliary datasets (Fig. 5b), we see that the former is slightly lower, due to that the main datasets have higher complexity, which can better reflect the performance on real datasets than simulated datasets from existing simulators that have the GRN effect only.

Despite that some of these methods perform better than others, their overall AUPRC ratio is far from satisfying even when true counts are used, indicating that inferring GRN from single-cell gene expression data is a challenging problem, and this can partly be due to that other factors, like chromatin accessibility, is not considered by this methods. We then investigated the performance of the 2 *multi-omics GRN inference* methods (scMTNI and CellOracle) that consider chromatin accessibility information. The multi-omics GRN inference methods first learn a prior GRN network from scATAC-seq or bulk ATAC-seq data, together with an input “TF by motif” matrix and “region by gene” matrix (Methods). scMultiSim provides ground truth “TF by motif” and “region by gene” matrices, but since these matrices obtained in practice are often noisy, we applied different levels of noise to the ground truth “TF by motif” and “region by gene” matrices and investigated the performance of CellOracle and scMTNI under four noise levels (0, 0.01, 0.1, 0.5) (Methods, “CellOracle_0/0.01/0.1/0.5” and “scMTNI_0/0.01/0.1/0.5” in Figs. 5a-b and S6).

As expected, the performances of both multi-omics methods decrease with larger noise levels. When comparing their performances with those of the 11 expression-based GRN inference methods, we see that the *multi-omics* methods have improved AUPRC ratio and AUROC values (Figs. 5a,b), supporting that the chromatin accessibility data provides valuable information for GRN inference, with the four noise levels on the “TF by motif” and “region by gene” matrices used in the test. When the noise level in the “TF by motif” and “region by gene” matrices in real data is very high, however, the improvement of the multi-omics methods can be limited. We also noticed that scMTNI has higher AUROCs but lower AUPRC ratios compared to CellOracle (Figs. 5, S6). The result is due to the high sparsity of the networks, where AUPRC ratio better reflects the true inference accuracy in the case of unbalanced data. Overall, these results demonstrated scMultiSim's broad applicability in benchmarking various computational methods that involve GRNs, especially the multi-omics GRN inference methods which have shown to be promising.

Benchmarking CCI inference methods

Spatially resolved single cell gene expression data provides a powerful tool for understanding cellular processes, tissue organization, and disease mechanisms at the single cell level. Methods have been proposed to infer CCIs using single cell gene expression data either with or without the spatial location information [3; 1; 9]. Methods inferring CCI from scRNA-seq data (without spatial location information of cells) have been previously compared and evaluated [44; 19]. While the cell spatial location information is considered to greatly assist the inference of CCIs, the quantitative accuracy of such methods is largely unknown. SRTsim and mistyR can be used to evaluate methods that infer CCIs between cell types, but they can not test single-cell-level CCI inference methods, along with their respective limitations in their simulation discussed in previous sections.

Using scMultiSim, one can evaluate single-cell-level and cell-type-level CCI inference methods that use spatially resolved gene expression data. We use both the main datasets and auxiliary datasets `C` and `S` (Table 2) to test the CCI inference methods, where the auxiliary datasets use different ground truth GRN and

CCI interactions (Fig. S7a) from the main datasets.

For cell-type-level methods, we tested Giotto [20], SpaOTsc [8] and SpaTalk [58]. Giotto and SpaTalk calculate the co-expression of ligand-receptor pairs between spatially-adjacent cell pairs, and conduct permutation test to measure the significance of their interaction across cell types. SpaOTsc learns CCIs by minimizing an optimal-transport-based loss function that models both spatial locations and expression values of ligand-receptor pairs. Since SpaTalk requires a minimum of 3 genes from the receptor to a downstream activated TF, which is not the case for the ground truth networks of the main datasets, we ran SpaTalk only on auxiliary dataset C.

We measured the inference accuracy using AUPRC and AUROC. When calculating the PRC and ROC curves, we applied different thresholds on Giotto's significance score and SpaTalk's Bonferroni corrected p-values. Fig. 5c and Fig. 5d respectively show the performance of these methods on the main and auxiliary datasets, while Fig. S7b shows ROC and PRC curves on auxiliary datasets and Fig. S8a shows the ROC curves, AUROC and AUPRC values on main datasets separated by cell population structures. Considering both AUROC and AUPRC, Giotto has the best overall performance. SpaTalk achieves comparable average AUROC with Giotto (Fig. 5d) but has a larger variation across runs compared to Giotto. Furthermore, SpaTalk outputs multiple identical p-values, causing a skewed distribution of the data points on the ROC and PRC curves (Fig. S7b).

In biology, CCIs between cell types are realized through CCIs between single cells. However, there are much less methods that infer single-cell-level CCIs compared to cell-type-level CCIs. We tested 2 methods that can infer single-cell-level CCIs, SpaOTsc and COMMOT [9]. Both methods use optimal transport frameworks, but SpaOTsc infers CCI of each ligand-receptor pair independently, and COMMOT also models the competing effects between different ligand-receptor pairs [9]. The advantage of modeling the competing effects is reflected in the superior performance of COMMOT (Fig. 5e and Fig. S8b; the calculation of AUPRC and AUROC is described in Methods). We note that the AUPRC values of both methods are low, and this is primarily due to the high sparseness in ground truth CCIs between single cells. Most of the AUPRC values are still higher than that of a random predictor which is 0.0012.

Benchmarking clustering and trajectory inference methods

scMultiSim can naturally be used to test methods for two classical problems: cell clustering and trajectory inference, using the scRNA-seq modality in our discrete main datasets (MD, Table 1). We tested five clustering methods, PCA-KMeans, CIDR [43], SC3 [37], TSCAN [32], and Seurat [27] (Fig. 6a). For each method and each dataset in the main datasets, we vary a key parameter "number of clusters" (if this parameter can directly be specified) or "resolution" (which indirectly controls the number of clusters). From Fig. 6a, all methods have the best performance when the cluster number is the ground truth value. In general, Seurat and SC3 have better performance than the others, which is consistent with previous benchmarking [21]. We also compared results on datasets with different σ_{cif} values (Fig. S9a-b), and observed that the methods generally have higher ARI with a smaller σ_{cif}, which is expected. Additionally, Seurat's recommended resolution range (0.4-1.2) provides an accurate estimation of the number of clusters (Fig. S9c).

We evaluated the performance of five trajectory inference methods (PAGA [69], Monocle [54], Slingshot [61], MST [56], pCreode [29]) on tree-structured trajectories using the `MT` datasets (Table 1). We calculated the R^2 and kNN purity (Methods) for each separate lineage in each dataset (Fig. 6b). Overall, PAGA Tree and Slingshot have the best performance, which is in line with previous benchmarking efforts [56; 74]. When comparing results on datasets with $\sigma_{\text{cif}} = 0.1$ and $\sigma_{\text{cif}} = 0.5$ (Fig. S10a-b), we again see that smaller σ_{cif} corresponds to better results.

Benchmarking RNA velocity estimation methods

Finally, we demonstrate scMultiSim's ability of benchmarking RNA velocity estimation methods by running two representative RNA velocity inference methods, scVelo [6] and VeloCyto [39], on the simulated data. We compare all three models in scVelo package, including the deterministic, stochastic, and dynamical models. The auxiliary dataset `V` (Table 2) was used, which contains 72 datasets of different numbers of cells and genes, with or without GRN. We evaluate the accuracy of inferred RNA velocity using the *cosine similarity* between the direction of inferred and ground truth velocity, where a higher score shows a better inference result (Methods).

From the result shown in Fig. 6c, scVelo's deterministic model has the highest cosine similarity score on all datasets. On the other hand, the dynamical model of scVelo, being considered a generalized model of scVelo-deterministic and VeloCyto, does not produce the best result. Gorin *et al* [26] discussed a potential reason that can harm the performance of the dynamical mode, which is the mismatch between the implicit assumption of dynamical scVelo and the true biological dynamics. In spite of the performance differences, the similarity scores are low (around 0.2) for all methods. We further calculated the cosine similarity with kNN smoothing (Methods), similar to the practice in [10], where the inferred RNA velocity of each cell is further averaged with the velocity of all its neighboring cells, and higher cosine similarity is achieved after the smoothing (Fig. 6d). We hypothesize that the regulatory information between genes can potentially improve the accuracy of inferred RNA velocity if this information is properly modeled. Therefore, we compared results on datasets with and without GRN effects, and found that presence of GRN effects does not have clearly improve the results (Fig. 6c-d). This can be due to that these methods do not explicitly take advantage of the regulatory relationships between genes, and future methods which do incorporate this information can use our simulated datasets for evaluation.

Discussion

We presented scMultiSim, a simulator of single cell multi-omics data which is able to incorporate biological factors including cell population, chromatin accessibility, RNA velocity, GRN and spatial CCIs to the output data. We demonstrated the presence of these simulated factors in the generated data, verified the relationship across modalities, and showcased the versatility of scMultiSim through benchmarking on various computational problems. Furthermore, by obtaining consistent benchmarking results with previous works [53; 21; 56], the simulated biological effects are validated to be practical and ready for real-world use.

Compared to existing simulators that mainly model one or two biological factors, scMultiSim generates data with more biological complexity similar to real data. This additional complexity enables researchers to better estimate the real-world performance of their methods on noisy experimental data, and allows for evaluating and applying methods for different computational tasks on the same datasets, which is the case when working with real datasets. Furthermore, with the coupled data modalities in the output, researchers can benchmark computational methods that use multiple modalities, which was previously infeasible.

scMultiSim's extensibility and versatility are central to its modular design, making it easy to include more biological factors and modalities in its simulations. For example, the framework used to model chromatin regions (RIV) and genes (GIV) can also be extended to include other data modalities, such as the protein abundance data. Additionally, we have shown that our CIF/GIV model is versatile enough to mathematically represent the effects of various biological mechanisms like GRNs and CCIs. In addition to the standard functions of scMultiSim, the model can be expanded to consider more realistic scenarios. For instance, the GRN can be set to a cell-specific and cell-type-specific mode, allowing for a more dynamical simulation of regulatory interactions. Moreover, the scATAC-seq data and scRNA-seq data can follow different trajectories or clustering structures, while the cell clusters can form less regular shapes than the current convex shapes.

scMultiSim's usability is supported by several key features. First, it requires minimal and easy-to-construct input. For example, users do not need to prepare a backbone for the trajectory to control the cell population; instead, only a plain R phylogenetic tree or a text file with the Newick format tree is needed. Second, the parameters of scMultiSim are transparent and self-explanatory. Since their effects on the simulated data are clear, users can set these parameters according to their needs and design their own *in silico* data, which is not constrained by the availability of real data. scMultiSim also provides default parameters and parameter sets that correspond to multiple real datasets.

Like most of the computational tools, scMultiSim has its own limitations. First, as a *de novo* simulator, scMultiSim simulates key biological mechanisms that are known to occur in the cells and are used as modelling basis in majority of computational methods. *De novo* simulators can not explicitly model all biological processes given existing knowledge and computational tractability. Second, the benchmarking of computational methods using *de novo* simulators needs to be complemented by tests on real data. Simulated data can provide a proof of concept which is a basic step in method development and help evaluate the performance of methods in controlled settings, and therefore has been used in major benchmarking efforts [53; 56; 45], where real datasets were also used to gain comprehensive assessments.

Nevertheless, we underline that scMultiSim is a significant step forward in *de novo* simulators for single cell genomics data, and its major advantage is its ability to encode various factors into a single versatile model, thus creating a comprehensive multi-modal simulator that can benchmark an unprecedented range of computational methods. More importantly, the coupled data modalities in the output jointly provide more information than a single modality alone, making it ideal for designing and benchmarking new methods on multi-omics data. We believe that scMultiSim has the potential to be a powerful tool for fostering the development of new computational methods for single-cell multi-omics and spatial data.

Methods

A. The Beta-Poisson model and intrinsic noise

The master equation of the kinetic model represents the steady state distribution of a gene's expression level given its kinetic parameters, k_{on}, k_{off}, and s [49]. The Beta-Poisson model was shown to be equivalent to the master equation [35] with faster calculation. The gene expression level x (which is also the mRNA count) can be sampled from the following distribution:

$$y = \text{Beta}(k_{on}, k_{off}) \tag{1}$$

$$x = \text{Poisson}(y \cdot s) \tag{2}$$

Using the above Beta-Poisson distribution to generate the gene expression level is one mode to obtain mRNA count for a gene in a cell. This works if we only need to generate the spliced mRNA counts. If users also need to generate unspliced mRNA counts and RNA velocity, the other mode, called the "full kinetic model" is used. The Beta-Poisson model is used by default when only generating spliced counts for lower running time.

The sampling process from the Beta-Poisson distribution to obtain x introduces intrinsic noise to the data, which corresponds to the intrinsic noise in real data caused by transcription burst. The theoretical mean of the kinetic model, which is $(\frac{k_{on}}{k_{on}+k_{off}} \cdot s)$, corresponds to the gene expression level of the gene with no intrinsic noise. We introduced parameter σ_i which controls the intrinsic noise by adjusting the weight between the random samples from the Poisson distribution and the theoretical mean:

$$x_{\sigma_i} = \sigma_i \cdot x + (1-\sigma_i) \cdot (\frac{k_{on}}{k_{on}+k_{off}} \cdot s) \tag{3}$$

The intrinsic noise in scRNA-seq data is hard to reduce in experiments due to the snapshot nature of scRNA-seq data. The parameter σ_i allows users to investigate the effect of intrinsic noise on the performance of the computational methods.

B. Cell Identity Factors (CIFs) and Gene Identity Vectors (GIVs)

The length of the CIF and GIV, denoted by n_{cif}, can be adjusted by the user. Overall, we have a $n_{\text{cell}} \times n_{\text{cif}}$ CIF matrix for each kinetic parameter (Fig. S1a), where each row is the CIF vector of a cell. Correspondingly, we also have the $n_{\text{cif}} \times n_{\text{gene}}$ Gene Identity Vectors (GIV) matrix, (Fig. S1b) where each column is linked to a gene, acting as the weight of the corresponding row in the CIF matrix, i.e. how strong the corresponding CIF can affect the gene. In short, CIF encodes the *cell identity*, while GIV encodes the *strength of biological effects*. Therefore, by multiplying the CIF and GIV matrix, we are able to get a $n_{\text{cell}} \times n_{\text{gene}}$ matrix, which is the desired kinetic parameter matrix with the cell and gene effects encoded. Each cell has three CIF vectors corresponding to the three kinetic parameters k_{on}, k_{off}, and s, and similarly for the GIV vectors (Fig. S1a-b).

C. diff-CIF generates user-controlled trajectories or clusters.

When generating data for cells from more than one cell type, the minimal user input of scMultiSim is the cell differentiation tree, which controls the cell types (for discrete population) or trajectories (for continuous population) in the output. The generated scRNA-seq and scATAC-seq data reflect the tree structure through the diff-CIF vectors. The diff-CIF vectors are generated as follows: starting from the root of the tree, a Gaussian random walk along the tree (Fig. 2a) is performed for each cell to generate the $n_{\text{diff-CIF}}$ dimension diff-CIF vector. Parameter σ_{cif} controls the standard deviation of the random walk, therefore a larger σ_{cif} will produce looser and noisier trajectory structures. Another parameter r_d is used to control the relative number of diff-CIF to non-diff-CIF. With a larger r_d, trajectories are clear and crisp in the output; with a smaller r_d, the trajectory is vague, and the shape of the cell population is more controlled by other factors like GRN. For a discrete population, only the cell types at the tree tips are used; then cells of each type are shifted by a Gaussian distribution, controlled by the same σ_{cif} parameter. Therefore, a smaller σ_{cif} will produce clearer cluster boundaries.

For a heterogeneous cell population, cells have different development stages and types. Users should input a cell differentiation tree where each node represents a cell type. The tree provides a backbone for the trajectory in the cell population. Each dimension of the diff-CIF vector is sampled along the tree via Gaussian random walk. First, cells start at the root of the tree; then for each dimension, the diff-CIF value for all cells $\mathbf{v}$ is

$$v_i = \sum_{t=1}^{i} q_t \text{ where } q_t = \mathcal{N}(0, \sigma_t). \tag{4}$$

σ_t is the distance along the tree between cell t and $t-1$. Alternatively, users can use an impulse model (using the implementation in SymSim). The lengths of the non-diff-CIF and diff-CIF vectors can be controlled by the user. More diff-CIFs will result in a more clear trajectory pattern in the cell population, which corresponds to the input tree. With very few diff-CIFs, the cell population is mainly controlled by the GRN.

D. tf-CIF and GIV encode the GRN effects

To encode GRN effect in the simulated single cell gene expression data, the GIVs and CIFs are designed to include a "TF part" (Fig. S1a). Cells are generated one by one along the given cell differentiation tree, where the expressions of the TFs in the t^{th} cell affect the gene expression of cell $t+1$. Formally, the i^{th} position of the TF part (corresponding to the i^{th} TF) of in the CIF of cell $t+1$ is calculated as:

$$\text{tf-CIF}_i^{(t+1)} = \frac{\mathbf{x}_i^{(t)}}{\mathbf{x}_i^{(t)} + \frac{1}{n}\sum_l \mathbf{x}_l^{(t)}} \qquad \forall i \in \text{TFs} \tag{5}$$

where $\mathbf{x}_i^{(t)}$ is the expression level of the i^{th} TF in the t^{th} cell. The corresponding tf-CIF for the root cell is sampled randomly from the Gaussian distribution $\mathcal{N}_{\text{cif}}$ supplied by the user.

The TF part of the GIV for a gene also has length of n_{TF}(Fig. S1b). Considering all genes, we have a $n_{\text{gene}} \times n_{\text{TF}}$ matrix, which we call the GRN effect matrix. This matrix encodes the ground truth GRN that is supplied by the user. Naturally, the GRN effect matrix is included in the GIV when calculating the s parameter, where the value at (i, j) is the regulation strength of TF j on gene i. Therefore, a larger regulation strength will lead to higher s, and consequently, higher expressions for the target genes. For k_{on} and k_{off}, the tf-CIF vector is sampled using $\mathcal{N}_{\text{cif}}$, assuming that the GRN does not affect the active state of a gene. However, in the scenario where it is desired to model GRN effect also in k_{on} and k_{off}, similar GRN effect matrix for s can be used for k_{on} and k_{off}.

scMultiSim also allows the use of ground truth GRNs which are cell specific. In this mode, random GRN edges are generated or deleted gradually along the pseudotime at a user-controlled speed. When simulating each cell, the tf-GIV will be filled with the current GRN effect matrix. The cell-specific GRN ground truth is outputted in this mode.

E. lig-CIF and GIV encode cell-cell interactions

When simulating spatial transcriptomics data with CCI effects, we used a 2-D $k \times k$ grid to model the spatial locations of cells (Fig. S1d). The grid size k is large enough to accommodate the n cells (can be specified by the user; if not provided, use 250% of cell number by default). A cell can have at most n_{nbs} neighbors with CCI (within the blue circle's range in Fig. 2a, and this radius can be adjusted). Therefore, the ligand CIF and GIV are of length $n_{\text{lig}} \cdot n_{\text{nbs}}$, where n_{lig} is the number of ligands.

The lig-GIV vector contains the CCI strength values, for example, the "n2 lg3" entry in Fig. 2a indicates the strength of CCI between the ligand 3 from the neighbor at position 2 and the receptor 2 of the central cell. The lig-CIF of each cell will inherit from its previous cell during the simulation process, which is similar to the tf-CIF mentioned above. Each entry of the lig-CIF vector corresponds to a ligand from one neighbor. The same Gaussian distribution $\mathcal{N}_{\text{cif}}$ is used for k_{on} and k_{off}. For s, due to the similarity of the ligand-receptor pairs and the TF-target pairs, we use a similar strategy as tf-CIF (shown in Eq. 5): cell i's lig-CIF is the normalized vector of cell $i-1$'s gene expression counts of the ligand genes (See Fig. 2a, Fig. S1).

To generate ground truth CCIs both at the cell type level and single cell level, scMultiSim pre-defines a ligand-receptor database, represented by a user input $m \times 3$ matrix S. There are m ligand-target pairs in total that correspond to each row of S. For each pair i, there are three parameters: ligand gene L_i, receptor gene T_i, and the effect E_i, representing how strongly the ligand can affect the expression of the receptor. For each cell type pair, the ground truth CCI beetween these two cell types are sampled from the ligand-receptor database (corresponds to the columns in S). For each neighboring cell pair, the ground truth CCIs between them follow the cell-type-level ground truth CCIs: if the two cells belong to two cell types C_1 and C_2 respectively (where C_1 can be the same as C_2), the CCIs between these two cells follow the CCIs defined in S corresponding to pair (C_1, C_2). Users can have further fine-grained control for each cell pair by letting it use a subset of ligand-receptor pairs sampled from the cell-type level ground truth.

F. Generating spatial data with different layouts and coupled with temporal processes

At each step t, a new cell is considered to be born and added to the grid. When adding a new cell, it has a probability of p_n (cell type affinity) to be a neighbor of an existing cell with the same cell type. Additionally, after the final time step, the user can choose to continue simulating t_c steps with all the cells in place (default is 10). This could increase the CCI effects in the final stable state.

We also provide other strategies to place a new cell, including (1) all cells placed at a random location, and (2) only the first m cells are randomly placed, and the remaining follow p_n. As different tissues can have different organizations of cell types in space, scMultiSim provides multiple options for the spatial layout of cells: “layers”, where cell types form layers in the tissue like in brain cortex; “islands”, like tumors surrounded by other cell types; “default”, that can result in a variation of layouts by tuning the p_n parameter.

Two additional layouts are provided: ”islands” and “layers”. When using these two layouts, the cell locations are pre-assigned before the simulation starts. For “islands”, the user can specify which cell types should form islands. We first generate the island shapes by randomly putting new cells next to an existing cell, then place the islands in the grid, and ensure they do not overlap. Next, the non-island cell types are placed in the grid randomly with a similar affinity parameter p_n. For “layers”, we still put a new cell next to an existing cell, but first sort all cells according to their type to generate the layer structure. Finally, a small proportion of all cells have their locations randomized to add some noise.

A pre-defined cell differentiation tree is required as input to define the differentiation topology in the cells. A new cell will always be in the initial state at the root of the differential tree. At each step, an existing cell moves forwards along a random path in the cell differential tree, representing the cell development. The gene expressions in the final step are output as the observed data. The cells will have different developmental stages at the final step, i.e., located at different positions along the tree. Therefore, the final output will contain the trajectory defined by the tree. Fig. S1 shows the structure for the CCI mode.

Although we collect cells at the last time point as our output (which is the case for real data), different cell types are guaranteed to present in the last step since the cells are added at different time steps, therefore having different development stages. In addition, we let the same cell (at the same location) have the same diff-CIF across different time steps, so the trajectory encoded in the diff-CIF is preserved in the final step. A cell’s TF and ligand CIF for the current step is inherited from the previous one to make sure other factors stay the same.

We use the following steps to calculate the correlation between the expressions of neighboring cells in Fig. 3e. First, a specific ligand-receptor pair (l, r) is chosen. Let $T(a, b) = \{\text{true}, \text{false}\}$ denote that there is CCI between cell a and cell b for (l, r). Then, for each cell i, we get its neighbor list n_i, which is a vector of 4 cells. A vector of 4 non-adjacent cells m_i is also randomly sampled for this cell. Thus, let x_c^g denote the gene expression of cell c and gene g. we calculate the "neighbor cells with CCI" correlation using the pairs $\{(x_i^l, x_j^r) | j \in n_i, T(i, j) = \text{true}\}$, the "neighbor cells without CCI" correlation using the pairs $\{(x_i^l, x_j^r) | j \in n_i, T(i, j) = \text{false}\}$, and the "non-neighbor cells" correlation using the pairs $\{(x_i^l, x_j^r)|j \in m_i\}$. Cell pairs of the same type are ignored while calculating the correlations because they tend to have similar expressions.

User can select to enable outputting the single-cell-level CCI ground truth. When enabled, the interacting neighbor cell pairs are pre-determined at the beginning by sampling 80% of the interacting edges from the cell-type-level ground truth.

G. Generating the Gene Identity Vectors

A gene's GIV vector has the same length as the CIF vectors. The values in the GIV of a gene act as the weights of the corresponding factors in the CIF, *i.e.*, how strong the corresponding CIF can affect the gene (Fig. 2a). If we have n_{gene} genes, we obtain a GIV matrix of size $n_{\text{cif}} \times n_{\text{gene}}$.

It can be divided into four submatrices as shown in Fig. S1b. For k_{on} and k_{off}, the non-diff-CIF and diff-CIF are sampled from distribution $\mathcal{G}$ as shown below:

$$\begin{cases} \mathcal{N}_{\text{giv}} & \text{w.p. } 1 - p_0^{\mathcal{G}} \\ 0 & \text{w.p. } p_0^{\mathcal{G}} \end{cases} \tag{6}$$

where $p_0^{\mathcal{G}}$ is a parameter specifying the probability of being zero, and $\mathcal{N}_{\text{giv}}$ is a user-adjustible Gaussian distribution. tf-GIV and lig-GIV are all zeros since TF/ligands affect s only. For s, the tf-GIV submatrix is the GRN effect matrix, i.e. a $n_{\text{TF}} \times n_{\text{gene}}$ matrix where the entry at (i, j) is the regulation effect between TF i and gene j. Similarly, the lig-GIV submatrix is the cell-cell interaction effect matrix. The nd-GIV submatrix is sampled from $\mathcal{G}$. For diff-GIV, we do the following steps to incorporate the connection between TFs and regulated genes: (1) Randomly select 2 GIV entries for each TF gene and give them a fixed small number. (2) For every target gene, it should use the same GIV vector as its regulators. If a gene has multiple regulators, its gene effects will be the combination of that of the regulators. This is achieved by multiplying the $n_{\text{diff}} \times n_{\text{TF}}$ GIV matrix in (1) and the $n_{\text{TF}} \times n_{\text{gene}}$ effect matrix. If a gene is both a TF and target, its GIV will be $0.5 \cdot ((1) + (2))$.

H. Simulating scATAC-seq data and relationship between scATAC-seq and scRNA-seq

Since scMultiSim incorporates the effect of chromatin accessibility in the gene expressions, the scATAC-seq data is simulated before the scRNA-seq data. The cell types in the scATAC-seq data can follow the same differentiation tree as in the scRNA-seq data (the scATAC-seq and scRNA-seq data share the same cells) or can follow a different tree (when the user want to impose controlled differences between modalities).

Similar to GIV, we use a randomly sampled *Region Identity Vector (RIV)* matrix to represent the chromatin regions. Following the same mechanism, we multiply the CIF and RIV matrix, and obtained a “non-realistic scATAC-seq” data matrix. Next, the scATAC-seq data matrix is obtained by scaling the “non-realistic” scATAC-seq data to match a real distribution learned from real data. This is a step to capture the intrinsic variation of the chromatin accessibility pattern, which we will also apply to the kinetic parameters when generating gene expressions.

The RIV matrix is sampled from a distribution $\mathcal{R}$ similar to $\mathcal{G}$:

$$\begin{cases} \mathcal{N}_{\text{riv}} & \text{w.p. } 1-p_0^{\mathcal{R}} \\ 0 & \text{w.p. } p_0^{\mathcal{R}} \end{cases} \tag{7}$$

where $p_0^{\mathcal{R}}$ is the probability of being zero and $\mathcal{N}_{\text{riv}}$ is a user-adjustable Gaussian distribution. With the CIF and RIV matrices, the $n_{\text{cell}} \times n_{\text{region}}$ scATAC-seq can be generated by (1) multiplying the CIF matrix by the RIV matrix, (2) scale the matrix to match the real data distribution, and (3) adding intrinsic noise (sampled from a small Gaussian) to the scATAC-seq data. In Step (2), we use the same rank-based scaling process as used for the kinetic parameters as described in Section "Preparing the kinetic parameters" above, and the real scATAC-seq data distribution is obtained from the dataset in [14].

To incorporate the relationship between scATAC-seq and scRNA-seq data, we use the scATAC-seq data to adjust the k_{on} parameter that is used to generate the scRNA-seq data, considering that chromatin accessibility affects the activated status of genes. Given matrix M_{tg} representing the TF-gene regulation (GRN), a TF motif to region matrix M_{tr}, and peak-to-gene matrix M_{rg} (Fig.1b), we consider the following relationship to be true:

$$M_{tg} = M_{tr} \cdot M_{rg} \tag{8}$$

Therefore, if the user inputs M_{tr} and M_{rg} directly, scMultiSim will generate the GRN according to Eq. 8. However, if the user inputs the GRN (M_{tg}) instead, scMultiSim can generate the other two matrices automatically. First, the region-to-gene M_{rg} matrix is generated to represent the mapping between chromatin regions and genes, where a gene can be regulated by 1-3 consecutive regions. Users can input a region distribution vector $\mathbf{r}$, for example, $\mathbf{r} = (0.1, 0.5, 0.4)$ means a gene can be regulated by three regions, and the probability of it being regulated by one, two and three consecutive regions are 0.1, 0.5 and 0.4, respectively. Then, the binary motif-to-region matrix is constructed by setting $M_{tr}^{(i,j)} = 1$ if a TF j regulates any gene corresponding to region i. The scATAC-seq data is also used to adjust k_{on} as described in the following section.

I. Preparing the kinetic parameters

The kinetic parameters, k_{on}, k_{off} and s are needed when generating single cell gene expression data (mRNA counts) using the kinetic model or Beta-Poisson distribution (Fig. 1b). While the basic idea is to get the parameter matrix using CIFs and GIVs (Fig. 1b), the three parameters go through different post-processing after the step of CIF × GIV. We first denote the result of CIF × GIV for k_{on}, k_{off} and s as M_1, M_2 and M_3, respectively.

(i) k_{on}. Since chromatin accessibility controls the activation of the genes, the scATAC-seq data is expected to affect the k_{on} parameter. We first prepare a $n_{\text{region}} \times n_{\text{gene}}$ 0-1 region-to-gene matrix Z using $\mathbf{r}$, where Z_{ij} indicates region i is associated with gene j (Z is outputted as the region-to-gene matrix). We multiply the scATAC-seq matrix with Z to get the $n_{\text{cell}} \times n_{\text{gene}}$ parameter matrix M_1'. Since the scATAC-seq data is sparse, there are many zeros in M_1'. Thus, we replace the zero entries in M_1' with the corresponding entries in M_1 (scaled to be smaller than the smallest non-zero entry in M_1') to help differentiate the zero entries. Finally, M_1' is sampled to match the distribution of k_{on} inferred from real data.

(ii) k_{off}. The parameters are obtained by scaling M_2 to match the real data distribution. For both k_{on} and k_{off}, it is possible to adjust the bimodality of gene expressions [74] through an optional bimodal factor B. A larger B will downscale both k_{on} and k_{off}, therefore increasing the bimodality.

(iii) s. The parameters are obtained by scaling M_3 to match the distribution of s inferred from real data. Then, users can also use a “scale.s” parameter to linearly scale s. It allows us to adjust the size of cells – some datasets may tend to large cells and some tend to have small cells depending on the cell types being profiled.

When scaling a matrix (M_1', M_2, or M_3) to match a reference distribution (eg. the distributions of k_{on}, k_{off} and s estimated from real data), the procedure is as follows: denoting the reference distribution by $\mathcal{D}$, the matrix to rescale by X, and the number of elements in X by n, we sample n ordered values from $\mathcal{D}$, then replace the data in X using the same order. scMultiSim uses the reference kinetic distribution parameters provided in SymSim [74], where the kinetic parameters are estimated from real data via an MCMC approach. The data used are the UMI-based dataset of 3005 cortex cells by Zeisel et al. [72], and a non-UMI-based dataset of 130 IL17-expressing T helper cells (Th17) by Gaublomme et al [24].

J. Generating RNA velocity with the full kinetic model

When using the full kinetic model, scMultiSim can generate the spliced and unspliced counts for each cell from the kinetic parameters. The starting spliced count x_s and unspliced count x_u for a cell are the previous cell's counts on the differential tree. For the first cell, the spliced/unspliced counts are

$$x_s = \frac{s \cdot k_{on} \cdot \beta}{k_{on} + k_{off}} \qquad x_u = \frac{s \cdot k_{on} \cdot d}{k_{on} + k_{off}} \tag{9}$$

where β and d respectively represent the splicing and degradation rate of genes. Both γ and d are sampled from a user-controlled normal distribution.

We set the cell cycle length to be $L = \frac{1}{k_{on}} + \frac{1}{k_{off}}$, and divide it into multiple steps. The number of steps follows $m = \left\lceil \frac{L}{\min(1/k_{on}, 1/k_{off})} \right\rceil$. We also provide an optional cell length factor η_L parameter to scale the cycle length. The probabilities of gene switching on or off are then calculated with $p_{\text{on}} = \frac{k_{on}}{m \cdot L}$ and $p_{\text{off}} = \frac{k_{off}}{m \cdot L}$. In each simulation step, we update the cell's current on/off state based on p_{on} and p_{off}, and generate the spliced/unspliced counts x_s and x_u. The spliced counts at step t are obtained by:

$$x_s^t = x_s^{t-1} + \frac{L}{m}(\beta \cdot x_u^{t-1} - d \cdot x_s^{t-1}) \tag{10}$$

and the unspliced counts are obtained by:

$$x_u^t = \begin{cases} x_u^{t-1} + \frac{L}{m}(s - \beta \cdot x_u^{t-1}) & \text{if state is on} \\ x_u^{t-1} - \frac{L}{m}(\beta \cdot x_u^{t-1}) & \text{if state is off} \end{cases} \quad \textbf{(11)}$$

The outputted x_s and x_u are the values at the final step $t = m$. The ground truth RNA velocity is calculated as:

$$v = \beta \cdot x_u - d \cdot x_s \quad \textbf{(12)}$$

When benchmarking the computational methods, we obtain the KNN averaged RNA velocity by applying a Gaussian Kernel KNN on the raw velocity data, with $k = \lceil n_{\text{cell}}/50 \rceil$. Then we normalize the velocity by calculating each cell's normalization factor $s_i = |v_i|$, where v_i is the velocity vector for cell i.

K. Adding technical noise and batch effects to data

Technical noise is added to the true mRNA counts to generate observed counts (observed scRNA-seq data) (Fig. 1b). The workflow follows SymSim's approach [74]: we simulate multiple rounds of mRNA capture and PCR amplification, then sequencing and profiling with UMI or non-UMI protocols. The parameter α controls the capture efficiency, that is, the rate of subsampling of transcripts during the capture step, which can vary in different cells, and user can specify it using a Normal distribution $\alpha \sim \mathcal{N}(\alpha_\mu, \alpha_\sigma)$. The sequencing depth $d \sim \mathcal{N}(d_\mu, d_\sigma)$ is another parameter that controls the quality of the observed data.

Batch effects are added by first dividing the cells into batches, then adding gene-specific and batch-specific Gaussian noise based on shift factors. For each gene j in batch i, the shift factor is sampled from Unif$(\mu_j - e_b, \mu_j + e_b)$, where $\mu_j \sim \mathcal{N}(0,1)$, and e_b is the parameter controlling the strength of batch effects. We provide several settings for adding highly expressed genes to help researchers fit the housekeeping genes in real data. scMultiSim also supports cell- and gene-wise tuning of the mRNA capture efficiency during the PCR process; therefore per-cell and per-gene metrics (such as zero count proportion and count variance) in the observed data can be controlled separately.

For scATAC-seq data, as the data is sampled from real data we do not explicitly simulate the experimental steps. We do provide methods to add batch effects to obtain multiple batches of scATAC-seq data.

L. Comparing statistical properties of simulated data with experimental data

To measure scMultiSim's ability to generate realistic data while incorporating all the effects, we compare the statistical properties of a real mouse somatosensory cortex seqFISH+ [23] dataset with simulated data generated using selected parameters. The dataset, with 10000 genes and spatial locations of 523 cells, is featured in Giotto [20]'s tutorial.

The scMultiSim simulated data has both GRN and CCI effects. The GRN used as input to scMultiSim is obtained as follows: GENIE3 [31] was used to obtain an inferred GRN from the dataset, then after looking at the output edge importance values, the top 200 edges were utilized to form a reference GRN. We used this GRN (96 genes) and another randomly sampled 104 genes to generate a subsample of the data. We then simulated a dataset with 200 genes and 523 cells using scMultiSim. After observing the dimension reduction of the real dataset, a discrete cell population is assumed. We specify the cluster ground truth using the exact cell type labels in the dataset. There are 10 cell types in total. We also used Giotto [20] to infer the cell-cell interactions between cells. We chose the top-seven most significant ligand-receptor pairs from Giotto's output, with p-value ≤ 0.01, more than 10 ligand and 10 receptor cells, and the largest `log2fc` values. For the ATAC-seq data, we set the "atac.p_zero" parameter to the empirical zero count proportion and, and the "atac.density" to the density of the reference dataset.

We used dyngen [10] as a baseline simulator to compare with scMultiSim. We generated a simulated dataset with dyngen, using the same GRN and number of cells. The cell types and cell-cell interaction ground truth were not provided since dyngen does not support them. Yet, we supplied the raw mouse SS cortex count matrix to dyngen's `experiment_params` as a reference dataset.

We used the following metrics to compare the distribution of simulated and experimental gene expression data, which is also used in [18]: library size (per cell), zero counts proportion (per cell), zero counts proportion (per gene), mean counts (per gene), counts variance (per gene), and the relationship between zero counts and mean counts per gene. We also used the following metrics for chromatin accessibility data as seen in [50; 41]: library size (per cell), cell sparsity (zero ATAC counts proportion per cell), and peak mean (mean ATAC counts per peak).

We use the same process for the other three datasets shown in S4. The datasets are 10x Multinome PBMC 3k (https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-no-cell-sorting-3-k-1-standard-2-0-0), ISSAC-seq Mouse Cortex [70], and MERFISH Mouse Hypothalamus [48].

M. Evaluation metrics for benchmarking computational methods

When evaluating the trajectory inference methods, we calculate the coefficient of determination R^2 and the kNN purity for all cells on each lineage. Given the cells' ground truth pseudotime vector t and the inferred pseudotime $\hat{t}$, the R^2 is equal to the square of the Pearson correlation coefficient:

$$R^2 = 1 - \frac{\sum_i (t_i - \hat{t}_i)^2}{\sum_i (t_i - \bar{t})^2} = \rho^2(t, \hat{t}) \tag{13}$$

where $\bar{t}$ is the mean of t. Given a cell i's kNN neighborhood $N_i^{\hat{t}}$ in $\hat{t}$ and its kNN neighborhood N_i^t in t, the kNN purity K_p for the cell is the Jaccard Index of N_i^t and $N_i^{\hat{t}}$.

The evaluation metrics used for multi-model data integration methods, Graph Connectivity and ASW, are described as following.

Graph Connectivity is defined as:

$$GC = \frac{1}{|C|}\sum_{c\in C}\frac{|LCC(c)|}{|c|} \tag{14}$$

where C is all cell types, $LCC(c)$ is in the largest connected component for cells of type c.

For the ASW:

$$batch\,ASW = \frac{1}{|M|}\sum_{k\in M}\frac{1}{|C_j|}\sum_{i\in C_j} 1-|silhouette(i)| \tag{15}$$

where M is the set of all cell types, and C_j is all the cells of type j. We used the implementation in [45].

When evaluating RNA velocity inference methods, we used the *cosine similarity* between the averaged estimated velocity and the ground truth. Calculating the average of estimated velocity vectors is commonly used to reduce local noise [10]. In dyngen [10], averaged RNA velocities were calculated across cells at trajectory waypoints weighted through a Gaussian kernel using ground truth trajectory; while in scMultiSim, we averaged the raw velocity values by kNN with a Gaussian kernel and $k = n_{\text{cells}}/50$ to achieve a similar averaging effect. Finally, cosine similarity is calculated as:

$$\frac{1}{n_{\text{cells}}}\sum_i \frac{v_i \cdot u_i}{\|v_i\|\|u_i\|} \tag{16}$$

where v_i is the ground truth velocity vector for cell i, and u_i is the predicted velocity vector.

N. Details on running data integration methods

We used all 144 main datasets. Technical noise and batch effects were added using default parameters (non-UMI, $\alpha \sim \mathcal{N}(0.1, 0.02)$, depth $\sim \mathcal{N}(10^5, 3000)$, ATAC observation probability 0.3). All integration methods were run on the scRNA and scATAC data with technical noise and batch effect. For Seurat-bridge, we followed the vignette "Dictionary Learning for cross-modality integration" in Seurat 4.1.0 using the default parameters. For UINMF, we used the latest GitHub release. We followed the "UINMF integration of Dual-omics data" tutorial and ran the `optimizeALS` method using $k = 12$. For Cobolt, we used the GitHub version cd8015b, with 10 latent dimensions, learning rate 0.005. If the loss diverged, we automatically retry with learning rate 0.001. The metrics, including ARI, NMI, Graph Connectivity, and ASW were computed using the scib [45] package.

O. Details on running GRN inference methods

The dataset `G` (Table 2) was generated using the following configurations: 100-gene GRN in Fig. 3[18], 1000 cells, 50 CIFs, $r_d = 0.2$, $\sigma_i = 1$, and other default parameters. We generated a total of 8 datasets `G` with random seed from 1 to 8. Technical noise and batch effect were then added using default parameters.

When testing the *expression-based GRN inference* methods, we used BEELINE's benchmark workflow [53] (GitHub version *79775f0*) and inferred GRNs from (1) true counts in the 8 datasets, and (2) observed counts with batch effects in the 8 datasets. We tested a total of 11 *expression-based GRN inference* methods, including PIDC, GRNBoost2, GENIE3, Sincerities, PPCOR, LEAP, GRISLI, SINGE, GRNVBEM, Scribe and SCODE. We measure the inference accuracy using AUPRC ratio and AUROC.

Both CellOracle and scMTNI take as input a noisy "GRN" learned from scATAC-seq data: peaks in scATAC-seq data can be assigned to different target genes according to their relative distance to the gene body on the genome. In scMultiSim, we can obtain the peak-to-gene matrix M_{rg} (also termed as gene activity matrix), and also the binary peak-by-TF matrix M_{tr}, as described before. The ground truth GRN is obtained by multiplying the gene activity matrix and peak-by-TF matrix. In reality, both gene activity matrix and peak-by-TF matrix obtained from scATAC-seq are highly noisy, where a lot of false positive connections exist. Thus, in scMultiSim, we randomly added connections in both matrices with probability $scale \times p$, and then randomly removed connections with probability p. p is the noise level parameter, and $scale$ controls the sparsity of both matrices after adding the noise. In our benchmarking, $scale$ is set to $50 \times p_{\text{pos}}$, where p_{pos} is the proportion of positive edges within each matrix. After applying noise to both matrices, we multiply them together to obtain the noisy "GRN". We ran CellOracle and scMTNI with noisy "GRN"s of different p's ($p = 0, 0.01, 0.1, 0.5$) to test how the noise affects the final inference accuracy. Both scMTNI and CellOracle infer cell-type level GRNs, to make the inferences result comparable with other population-level GRN inference methods, we set the number of clusters to be 1 in both methods. We then ran scMTNI and CellOracle following their tutorial using the same hyperparameter setting (scMTNI: `https://github.com/Roy-lab/scMTNI`, CellOracle: `https://morris-lab.github.io/CellOracle.documentation/notebooks/04_Network_analysis/Network_analysis_with_Paul_etal_2015_data.html`).

P. Details on running CCI inference methods

Apart from the main datasets, we generated 12 `C` datasets (Table 2) using the following procedure. for each dataset , we first construct the GRN (Fig. S7a): (1) let genes 1-6 be the transcription factors. Sample 70 edges from gene 1-6 to gene 7-53. (2) Connect gene 7-53 (regulator) to gene 54-100 (target) consecutively. (3) Connect gene 54-100 to gene 110-156. In this way, we can generate a GRN with reasonable edge density and make sure that there are three downstream genes for each TF, which is required by SpaTalk. Then we construct the ligand-receptor pairs: let the ligands be gene 101-106 and receptors be gene 2, 6, 10, 8, 20, and 30. We divide a linear trajectory into 5 sections, corresponding to 5 cell types. Between each cell type pair (excluding same-type pairs), we sample 3-6 ligand-receptor pairs and enable cell-cell interactions with them for the two cell types. The dataset is then simulated using 160 genes in total, 500 cells, and 50 CIFs. We also generated 8 `S` datasets using the same GRN and CCI network, but with 120 genes, 400 cells, and single-cell CCI ground truth enabled. We use the true counts to benchmark the methods.

To run SpaTalk, as suggested by the authors, we used the latest GitHub version and modified the original `plot_lrpair_vln` method to return the p-value from the Wilcoxon rank sum test directly, rather than drawing a figure. Before using the p-values to calculate the precision and recall, we adjusted them using Bonferroni correction:

$$\hat{p}_i = \max(p_i \cdot |p|, 1) \tag{17}$$

where p is the p-value vector for all cell types and ligand-receptor pairs. For Giotto, we used the R package 1.1.2 and followed the `mini_seqfish` vignette. For SpaOTsc, we used version 0.2 with default parameters. We ran the `spatial_signaling_ot` function for each ligand-receptor pair in the predefined list to get the single-cell level prediction. We then averaged the probability for all cells within each cell type to get the cell-type-level result.

COMMOT takes as input a predefined ligand-receptor database, and ranks the ligand-receptor pairs in the database according to the possibility of their interactions in each cell pair. We used all ligand-receptor pairs in the simulation ground truth as the input ligand-receptor database of COMMOT. In addition, COMMOT uses hyperparameter d_{thr} to control the interaction radius between ligands and receptors in the space. When running COMMOT, we set d_{thr} to be the maximum distances of cells to their 20th nearest neighbors.

COMMOT and SpaOTsc are able to infer cell-level ligand-receptor interactions. We formulated their inference results and the ground truth separately into 3D interaction tensors (cell by cell by ligand-receptor pair), where each entry corresponds to the interaction strength of a certain ligand-receptor pair between a cell pair. We then measured the AUROC and AUROC between the inferred interaction tensor and the ground truth interaction tensor as the benchmarking scores.

Q. Details on running clustering methods

We used CIDR 0.1.5, SC3 1.24.0, Seurat 4.1, and TSCAN 2.0. The parameters we specified are (1) SC3: `pct_dropout` $= [0, 100]$, (2) Seurat: `dims.use` $= 30$. For PCA-Kmeans, we simply ran Kmeans clustering on the first 20 principle components using the default R implementation `prcomp` and `kmeans`. ARI is calculated by `adjustedRandIndex` from the R package `mclust`. Some code was adapted from [21].

R. Details on running trajectory inference methods

We used the latest dynverse [56] package (dyno 0.1.2) to run the trajectory inference methods. When running them, we provide the correct root cell ID, number of starting clusters and number of ending clusters. The R^2 values are calculated between the inferred pseudotime and the ground truth for each separate lineage. The kNN purity value is calculated for each lineage as: for cell i, we obtain its k Nearest Neighbors N_i on the pseudotime with $k = 50$. Then the kNN purity for i is the Jaccard Index of N_i on the inferred pseudotime and N_i on the true pseudotime. R^2 measures the correctness of inferred pseudotime, but when there are multiple branches in the trajectory, R^2 does not distinguish cells with similar pseudotime but are on different branches. In this case, the kNN purity serves as a complementary measurement that measures the correctness of inferred trajectory backbone.

S. Details on running RNA velocity estimation methods

We use the datasets `V` to benchmark RNA velocity inference methods as shown in Table 2. We used scVelo 0.2.4 and VeloCyto 0.17.17. We benchmarked scVelo with three modes: `deterministic`, `stochastic`, and `dynamical`. For VeloCyto, we used the default options.

Data and Code Availability

The scMultiSim R package is available at https://github.com/ZhangLabGT/scMultiSim/tree/main. The code for dataset generation and benckmarking is available at https://github.com/ZhangLabGT/scMultiSim_manuscript. The simulated datasets are available at https://www.dropbox.com/sh/sfkn5hweaejbrir/AAB9liDyL8QuXd7LAgUsihGfa?dl=0.

Author Contributions

X.Z. conceived the idea and X.C. contributed to the design of scMultiSim. H.L., Z.Z. and M.S. implemented the software package. H.L. performed validations and benchmarks. H.L., X.Z. and Z.Z. wrote the manuscript.

Acknowledgements

This work was supported by the US National Science Foundation DBI-2019771, DBI-2145736 and National Institutes of Health grant R35GM143070 (HL, ZZ, XZ), the Guangdong Basic and Applied Basic Research Foundation (2022B1515120077 to XC) and the Shenzhen Innovation Committee of Science and Technology (20220815094330001 to XC).

Competing Interests Statement

The authors declared no competing interest.

Bibliography

1. A. A. Almet, Z. Cang, S. Jin, and Q. Nie. The landscape of cell-cell communication through single-cell transcriptomics. *Curr Opin Syst Biol*, 26:12–23, June 2021.
2. R. Argelaguet, A. S. E. Cuomo, O. Stegle, and J. C. Marioni. Computational principles and challenges in single-cell data integration. *Nat. Biotechnol.*, pages 1–14, May 2021.
3. E. Armingol, A. Officer, O. Harismendy, and N. E. Lewis. Deciphering cell-cell interactions and communication from gene expression. *Nat. Rev. Genet.*, 22(2):71–88, Feb. 2021.
4. P. Badia-I-Mompel, L. Wessels, S. Müller-Dott, R. Trimbour, R. O. Ramirez Flores, R. Argelaguet, and J. Saez-Rodriguez. Gene regulatory network inference in the era of single-cell multi-omics. *Nat. Rev. Genet.*, June 2023.
5. G. Baruzzo, I. Patuzzi, and B. Di Camillo. SPARSim single cell: a count data simulator for scRNA-seq data. *Bioinformatics*, 36(5):1468–1475, Mar. 2020.
6. V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat. Biotechnol.*, Aug. 2020.
7. R. Browaeys, W. Saelens, and Y. Saeys. NicheNet: modeling intercellular communication by linking ligands to target genes. *Nat. Methods*, 17(2):159–162, Feb. 2020.
8. Z. Cang and Q. Nie. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nat. Commun.*, 11(1):2084, Apr. 2020.
9. Z. Cang, Y. Zhao, A. A. Almet, A. Stabell, R. Ramos, M. V. Plikus, S. X. Atwood, and Q. Nie. Screening cell-cell communication in spatial transcriptomics via collective optimal transport. *Nat. Methods*, Jan. 2023.
10. R. Cannoodt, W. Saelens, L. Deconinck, and Y. Saeys. Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. *Nature Communications*, 12(1):1–9, 2021.
11. J. Cao, D. A. Cusanovich, V. Ramani, D. Aghamirzaie, H. A. Pliner, A. J. Hill, R. M. Daza, J. L. McFaline-Figueroa, J. S. Packer, L. Christiansen, F. J. Steemers, A. C. Adey, C. Trapnell, and J. Shendure. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. *Science*, 361(6409):1380–1385, Sept. 2018.
12. T. E. Chan, M. P. H. Stumpf, and A. C. Babtie. Gene regulatory network inference from Single-Cell data using multivariate information measures. *Cell Syst*, 5(3):251–267.e3, Sept. 2017.
13. S. Chen, B. B. Lake, and K. Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat. Biotechnol.*, 37(12):1452–1457, Dec. 2019.
14. X. Chen, R. J. Miragaia, K. N. Natarajan, and S. A. Teichmann. A rapid and robust method for single cell chromatin accessibility profiling. *Nature Communications*, 9(1):5345, 2018.
15. H. L. Crowell, S. X. Morillo Leonardo, C. Soneson, and M. D. Robinson. The shaky foundations of simulating single-cell RNA sequencing data. *Genome Biol.*, 24(1):62, Mar. 2023.
16. H. L. Crowell, C. Soneson, P.-L. Germain, D. Calini, L. Collin, C. Raposo, D. Malhotra, and M. D. Robinson. muscat detects subpopulation-specific state transitions from multi-sample multi-condition single-cell transcriptomics data. *Nat. Commun.*, 11(1):6077, Nov. 2020.
17. A. Deshpande, L.-F. Chu, R. Stewart, and A. Gitter. Network inference with granger causality ensembles on single-cell transcriptomics. *Cell Rep.*, 38(6):110333, Feb. 2022.
18. P. Dibaeinia and S. Sinha. SERGIO: A Single-Cell expression simulator guided by gene regulatory networks. *Cell Syst*, Aug. 2020.
19. D. Dimitrov, D. Türei, M. Garrido-Rodriguez, P. L. Burmedi, J. S. Nagai, C. Boys, R. O. Ramirez Flores, H. Kim, B. Szalai, I. G. Costa, A. Valdeolivas, A. Dugourd, and J. Saez-Rodriguez. Comparison of methods and resources for cell-cell communication inference from single-cell RNA-Seq data. *Nat. Commun.*, 13(1):1–13, June 2022.

20. R. Dries, Q. Zhu, R. Dong, C. H. L. Eng, H. Li, K. Liu, Y. Fu, T. Zhao, A. Sarkar, F. Bao, R. E. George, N. Pierson, L. Cai, and G. C. Yuan. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome Biology*, 22, 2021.
21. A. Duò, M. Robinson, and C. Soneson. A systematic performance evaluation of clustering methods for single-cell rna-seq data. *F1000Research*, 7:1141, 11 2020.
22. M. Efremova and S. A. Teichmann. Computational methods for single-cell omics across modalities. *Nat. Methods*, 17(1):14–17, Jan. 2020.
23. C.-H. L. Eng, M. Lawson, Q. Zhu, R. Dries, N. Koulena, Y. Takei, J. Yun, C. Cronin, C. Karp, G.-C. Yuan, and L. Cai. Transcriptome-scale super-resolved imaging in tissues by RNA seqFISH. *Nature*, 568(7751):235–239, Apr. 2019.
24. J. T. Gaublomme, N. Yosef, Y. Lee, R. S. Gertner, L. V. Yang, C. Wu, P. P. Pandolfi, T. Mak, R. Satija, A. K. Shalek, V. K. Kuchroo, H. Park, and A. Regev. Single-Cell Genomics Unveils Critical Regulators of Th17 Cell Pathogenicity. *Cell*, 163(6):1400–1412, 2015.
25. B. Gong, Y. Zhou, and E. Purdom. Cobolt: integrative analysis of multimodal single-cell sequencing data. *Genome Biology*, 22(1):351, 2021.
26. G. Gorin, M. Fang, T. Chari, and L. Pachter. RNA velocity unraveled. *PLOS Computational Biology*, 18(9):e1010492, 2022.
27. Y. Hao, S. Hao, E. Andersen-Nissen, W. M. M. III, S. Zheng, A. Butler, M. J. Lee, A. J. Wilk, C. Darby, M. Zagar, P. Hoffman, M. Stoeckius, E. Papalexi, E. P. Mimitou, J. Jain, A. Srivastava, T. Stuart, L. B. Fleming, B. Yeung, A. J. Rogers, J. M. McElrath, C. A. Blish, R. Gottardo, P. Smibert, and R. Satija. Integrated analysis of multimodal single-cell data. *Cell*, 2021.
28. Y. Hao, T. Stuart, M. H. Kowalski, S. Choudhary, P. Hoffman, A. Hartman, A. Srivastava, G. Molla, S. Madad, C. Fernandez-Granda, and R. Satija. Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nat. Biotechnol.*, May 2023.
29. C. A. Herring, A. Banerjee, E. T. McKinley, A. J. Simmons, J. Ping, J. T. Roland, J. L. Franklin, Q. Liu, M. J. Gerdes, R. J. Coffey, and K. S. Lau. Unsupervised trajectory analysis of Single-Cell RNA-Seq and imaging data reveals alternative tuft cell origins in the gut. *Cell Syst*, 6(1):37–51.e9, Jan. 2018.
30. Y. Hu, T. Peng, L. Gao, and K. Tan. CytoTalk: De novo construction of signal transduction networks using single-cell transcriptomic data. *Sci Adv*, 7(16), Apr. 2021.
31. V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring Regulatory Networks from Expression Data Using Tree-Based Methods. *PLOS ONE*, 5(9):1–10, 2010.
32. Z. Ji and H. Ji. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. *Nucleic Acids Res.*, 44(13):e117, July 2016.
33. S. Jin, C. F. Guerrero-Juarez, L. Zhang, I. Chang, R. Ramos, C.-H. Kuan, P. Myung, M. V. Plikus, and Q. Nie. Inference and analysis of cell-cell communication using CellChat. *Nat. Commun.*, 12(1):1088, Feb. 2021.
34. K. Kamimoto, B. Stringa, C. M. Hoffmann, K. Jindal, L. Solnica-Krezel, and S. A. Morris. Dissecting cell identity via network inference and in silico gene perturbation. *Nature*, 614(7949):742–751, Feb. 2023.
35. J. Kim and J. C. Marioni. Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. 14(1):R7, 2013.
36. S. Kim. ppcor: An R package for a fast calculation to semi-partial correlation coefficients. *Commun Stat Appl Methods*, 22(6):665–674, Nov. 2015.
37. V. Y. Kiselev, K. Kirschner, M. T. Schaub, T. Andrews, A. Yiu, T. Chandra, K. N. Natarajan, W. Reik, M. Barahona, A. R. Green, and M. Hemberg. SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods*, 14(5):483–486, May 2017.

38. A. R. Kriebel and J. D. Welch. UINMF performs mosaic integration of single-cell multi-omic datasets using nonnegative matrix factorization. *Nature Communications*, 13(1):780, 2022.

39. G. La Manno, R. Soldatov, A. Zeisel, E. Braun, H. Hochgerner, V. Petukhov, K. Lidschreiber, M. E. Kastriti, P. Lönnerberg, A. Furlan, J. Fan, L. E. Borm, Z. Liu, D. van Bruggen, J. Guo, X. He, R. Barker, E. Sundström, G. Castelo-Branco, P. Cramer, I. Adameyko, S. Linnarsson, and P. V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, Aug. 2018.

40. B. Li, W. Zhang, C. Guo, H. Xu, L. Li, M. Fang, Y. Hu, X. Zhang, X. Yao, M. Tang, K. Liu, X. Zhao, J. Lin, L. Cheng, F. Chen, T. Xue, and K. Qu. Benchmarking spatial and single-cell transcriptomics integration methods for transcript distribution prediction and cell type deconvolution. *Nat. Methods*, May 2022.

41. C. Li, X. Chen, S. Chen, R. Jiang, and X. Zhang. simcas: an embedding-based method for simulating single-cell chromatin accessibility sequencing data. *Bioinformatics*, 39(8), 2023.

42. C. Li, M. Virgilio, K. L. Collins, and J. D. Welch. Single-cell multi-omic velocity infers dynamic and decoupled gene regulation. Research in Computational Molecular Biology, pages 297–299. Springer International Publishing.

43. P. Lin, M. Troup, and J. W. K. Ho. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-Seq data. *bioRxiv*, page 068775, Aug. 2016.

44. Z. Liu, D. Sun, and C. Wang. Evaluation of cell-cell interaction methods by integrating single-cell RNA sequencing data with spatial information. *Genome Biol.*, 23(1):218, Oct. 2022.

45. M. D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and F. J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, 19(1):41–50, 2022.

46. S. Ma, B. Zhang, L. M. LaFave, A. S. Earl, Z. Chiang, Y. Hu, J. Ding, A. Brack, V. K. Kartha, T. Tay, T. Law, C. Lareau, Y.-C. Hsu, A. Regev, and J. D. Buenrostro. Chromatin potential identified by shared Single-Cell profiling of RNA and chromatin. *Cell*, 183(4):1103–1116.e20, Nov. 2020.

47. T. Moerman, S. Aibar Santos, C. Bravo González-Blas, J. Simm, Y. Moreau, J. Aerts, and S. Aerts. GRNBoost2 and arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 35(12):2159–2161, June 2019.

48. J. R. Moffitt, D. Bambah-Mukku, S. W. Eichhorn, E. Vaughn, K. Shekhar, J. D. Perez, N. D. Rubinstein, J. Hao, A. Regev, C. Dulac, and X. Zhuang. Molecular, spatial, and functional single-cell profiling of the hypothalamic preoptic region. *Science*, 362(6416), 2018.

49. B. Munsky, G. Neuert, and A. van Oudenaarden. Using gene expression noise to understand gene regulation. *Science*, 336(6078):183–187, Apr. 2012.

50. Z. Navidi, L. Zhang, and B. Wang. simATAC: a single-cell ATAC-seq simulation framework. *Genome Biol.*, 22(1):74, Mar. 2021.

51. N. Papili Gao, S. M. M. Ud-Dean, O. Gandrillon, and R. Gunawan. SINCERITIES: inferring gene regulatory networks from time-stamped single cell transcriptional expression profiles. *Bioinformatics*, 34(2):258–266, Jan. 2018.

52. J. Peccoud and B. Ycart. Markovian modeling of gene-product synthesis. 48(2):222–234, Oct. 1995.

53. A. Pratapa, A. P. Jalihal, J. N. Law, A. Bharadwaj, and T. M. Murali. Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nat. Methods*, Jan. 2020.

54. X. Qiu, Q. Mao, Y. Tang, L. Wang, R. Chawla, H. A. Pliner, and C. Trapnell. Reversed graph embedding resolves complex single-cell trajectories. *Nat. Methods*, 14(10):979–982, Oct. 2017.

55. S. G. Rodriques, R. R. Stickels, A. Goeva, C. A. Martin, E. Murray, C. R. Vanderburg, J. Welch, L. M. Chen, F. Chen, and E. Z. Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434), 2019.

56. W. Saelens, R. Cannoodt, H. Todorov, and Y. Saeys. A comparison of single-cell trajectory inference methods. *Nat.*

Biotechnol., Apr. 2019.

57. S. Shah, E. Lubeck, W. Zhou, and L. Cai. In situ transcription profiling of single cells reveals spatial organization of cells in the mouse hippocampus. *Neuron*, 92(2):342–357, Oct. 2016.
58. X. Shao, C. Li, H. Yang, X. Lu, J. Liao, J. Qian, K. Wang, J. Cheng, P. Yang, H. Chen, X. Xu, and X. Fan. Knowledge-graph-based cell-cell communication inference for spatially resolved transcriptomic data with SpaTalk. *Nature Communications*, 13(1):4429, 2022.
59. D. Song, Q. Wang, G. Yan, T. Liu, T. Sun, and J. J. Li. scdesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nature Biotechnology*, 2023.
60. P. L. Ståhl, F. Salmén, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss, A. Mollbrink, S. Linnarsson, S. Codeluppi, Å. Borg, F. Pontén, P. I. Costea, P. Sahlén, J. Mulder, O. Bergmann, J. Lundeberg, and J. Frisén. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics, 2016.
61. K. Street, D. Risso, R. B. Fletcher, D. Das, J. Ngai, N. Yosef, E. Purdom, and S. Dudoit. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. *BMC Genomics*, 19(1):477, June 2018.
62. T. Stuart, A. Butler, P. Hoffman, C. Hafemeister, E. Papalexi, W. M. Mauck, 3rd, Y. Hao, M. Stoeckius, P. Smibert, and R. Satija. Comprehensive integration of Single-Cell data. *Cell*, 177(7):1888–1902.e21, June 2019.
63. T. Sun, D. Song, W. V. Li, and J. J. Li. scDesign2: a transparent simulator that generates high-fidelity single-cell gene expression count data with gene correlations captured. *Genome Biology*, 22(1):163, 2021.
64. J. Tanevski, R. O. Ramirez Flores, A. Gabor, D. Schapiro, and J. Saez-Rodriguez. Explainable multiview framework for dissecting spatial relationships from highly multiplexed data. *Genome Biology*, 23(97), 2022.
65. K. Vandereyken, A. Sifrim, B. Thienpont, and T. Voet. Methods and applications for single-cell and spatial multi-omics. *Nat. Rev. Genet.*, pages 1–22, Mar. 2023.
66. L. Wang, N. Trasanidis, T. Wu, G. Dong, M. Hu, D. E. Bauer, and L. Pinello. Dictys: dynamic gene regulatory network dissects developmental continuum with single-cell multiomics. *Nature Methods*, 2023.
67. X. Wang, W. E. Allen, M. A. Wright, E. L. Sylwestrak, N. Samusik, S. Vesuna, K. Evans, C. Liu, C. Ramakrishnan, J. Liu, G. P. Nolan, F.-A. Bava, and K. Deisseroth. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science*, 361(6400), July 2018.
68. J. D. Welch, V. Kozareva, A. Ferreira, C. Vanderburg, C. Martin, and E. Z. Macosko. Single-Cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*, 177(7):1873–1887.e17, June 2019.
69. F. A. Wolf, F. K. Hamey, M. Plass, J. Solana, J. S. Dahlin, B. Göttgens, N. Rajewsky, L. Simon, and F. J. Theis. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.*, 20(1):59, Mar. 2019.
70. W. Xu, W. Yang, Y. Zhang, Y. Chen, N. Hong, Q. Zhang, X. Wang, Y. Hu, K. Song, W. Jin, and X. Chen. Issaac-seq enables sensitive and flexible multimodal profiling of chromatin accessibility and gene expression in single cells. *Nature Methods*, 19(10):1243–1249, 2022.
71. L. Zappia and F. J. Theis. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biol.*, 22(1):301, Oct. 2021.
72. A. Zeisel, A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. L. Manno, A. Juréus, S. Marques, H. Munguba, L. He, C. Betsholtz, C. Rolny, G. Castelo-Branco, J. Hjerling-Leffler, and S. Linnarsson. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142, 2015.
73. S. Zhang, S. Pyne, S. Pietrzak, S. Halberg, S. G. McCalla, A. F. Siahpirani, R. Sridharan, and S. Roy. Inference of cell type-specific gene regulatory networks on cell lineages from single cell omic datasets. *Nature Communications*, 14(1), 2023.

74. X. Zhang, C. Xu, and N. Yosef. Simulating multiple faceted variability in single cell RNA sequencing. *Nat. Commun.*, 10(1):2611, June 2019.

75. Z. Zhang, C. Yang, and X. Zhang. scDART: integrating unmatched scRNA-seq and scATAC-seq data and learning cross-modality relationship simultaneously. *Genome Biology*, 23(1):139, 2022.

76. Z. Zhang and X. Zhang. VeloSim: Simulating single cell gene-expression and RNA velocity. *BioRxiv*, 2021.

77. J. Zhu, L. Shang, and X. Zhou. SRTsim: spatial pattern preserving simulations for spatially resolved transcriptomics. *Genome Biol.*, 24(1):39, Mar. 2023.

Tables

Name (#datasets)	Label	Population structure	σ_{cif}	Cells	Genes	Velo	GRN	CCI	Seed
Main (144)	**M**	**L: Continuous Linear (Phyla1)** **T: Continuous Tree (Phyla3)** **D: Discrete (Phyla5)**	0.1	500	**1: 110 genes** **2: 200 genes** **3: 500 genes**	F	GRN_100	T	**a: 1** **b: 2** **c: 3** **d: 4**
				800	**4: 110 genes** **5: 200 genes** **6: 500 genes**				
			0.5	500	**7: 110 genes** **8: 200 genes** **9: 500 genes**				
				800	**10: 110 genes** **11: 200 genes** **12: 500 genes**				

Table 1. The main dataset contains 144 datasets with varying trajectory, σ_{cif},number of cells and genes. For each parameter configuration, four datasets are generated using different random seeds. We number the datasets for easy referencing in the text: starting with the letter M,then a letter {L,T,D} specifying the trajectory; followed by a number 1-12 identifying the configuration of σ_{cif}, number of cells and genes; and last, a lowercase letter a-d indicating the random seed. For example, MD5c uses a discrete cell population, $\sigma_{\text{cif}} = 0.1$, 800 cells, 200 genes and random seed 3. Phyla1, Phyla3 and Phyla5 are the input tree structure used to generate the cell populations, and they are shown in Fig. 2b.

Name (#datasets)	Label	Population structure	σ_{cif}	Cells	Genes	Velo	GRN	CCI	Seed	Other Params
Auxiliary (162)	**A**	Tree (Phyla5) Discrete (5 clusters)	0.1, 0.5, 1	1000,3000	200,1000	F	GRN_100	F	1,2	$E_{\text{atac}} = 0.2, 0.5, 0.9$ $\sigma_i = 0.3, 1$ $r_d = 0.2, 0.8$
Velocity (72)	**V**	Tree (Phyla5)	0.1	500,750,1000	100,200,500	T	GRN_100 N/A	T	5-8	
Add'l Integration (72)	**I**	Phyla5,Discrete	0.1	1500,2250,3000	100,200,500	F	GRN_100,N/A	F	1-4	
Add'l GRN (16)	**G**	Linear	0.1	1000	110,500	F	GRN_100	F	1-8	
Add'l CCI (8)	**C**	Linear	0.1	500	200	F	Fig. S7	T	1-8	
Add'l Single-cell CCI (8)	**S**	Linear	0.1	400	120	F	GRN_100	T	1-8	
Realistic (1)	**R**	Discrete (10 clusters)	0.1	523	200	F	Inferred	F	1	

Table 2. The auxiliary dataset and other datasets used in supplemental information. "GRN_100" refers to the 100-gene GRN as shown in Fig. 3a. For dataset `A`, 18 of the 162 datasets are "large" datasets with 3000 cells and 100 genes. To save time and space, we fixed several parameters ($\sigma_i = 1$, $r_d = 0.8$ and $seed = 1$) for these datasets.

Figures

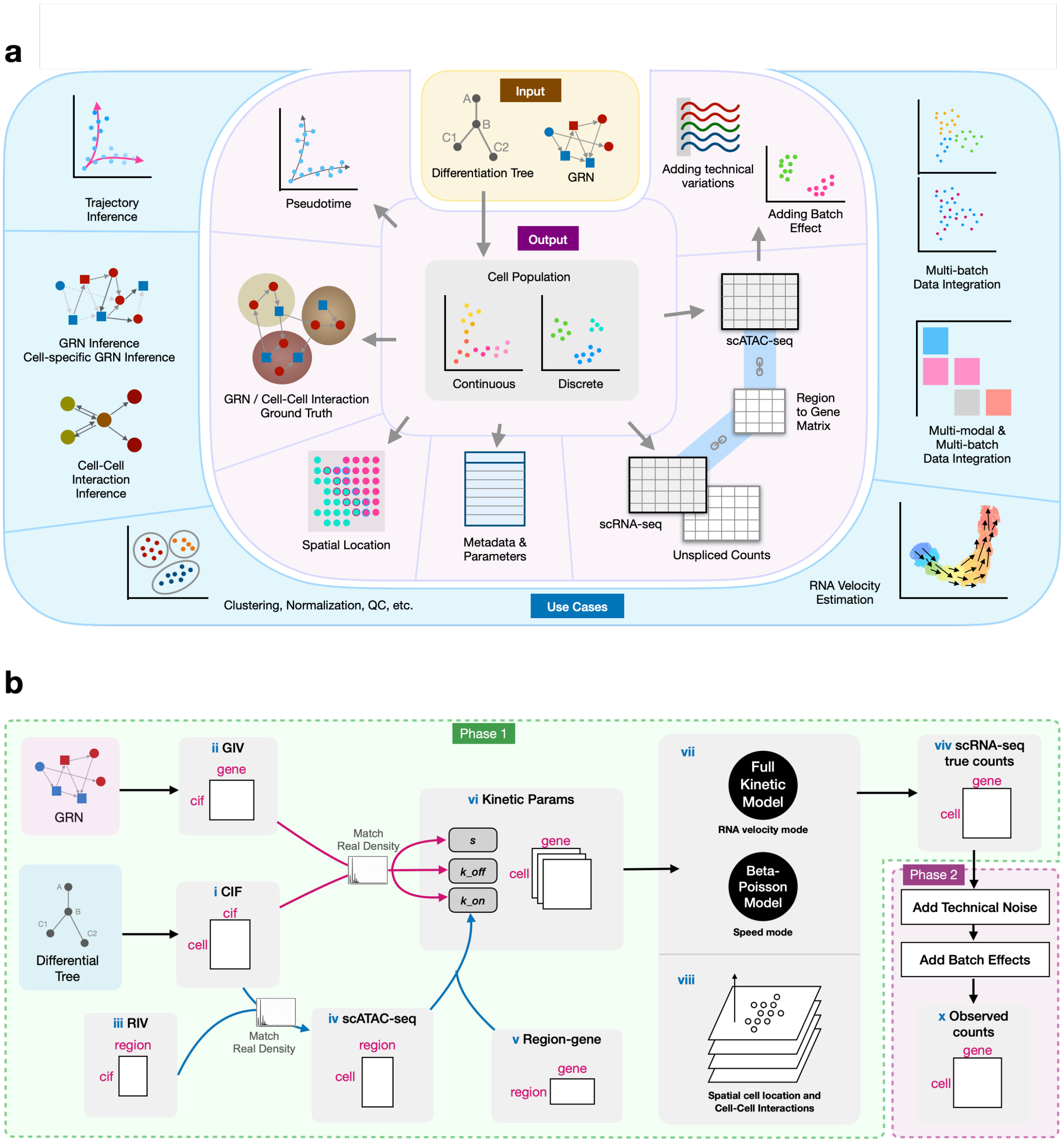

Figure 1. Overview of scMultiSim. See the next page for descriptions.

Figure 1. Overview of scMultiSim. (a): The input, output, and use cases. The minimal required input is a cell differential tree describing the differentiation relationship of cell types. It controls the cell trajectory or clusters in the output. A user-input ground truth GRN is recommended to guide the simulation. Users can also provide ground truth for cell-cell interaction and control each simulated biological effects using various parameters. **(b)**: The overall structure of scMultiSim. The scATAC-seq data (iv) is firstly generated using CIF (i) and RIV (iii). The kinetic parameters used to generate scRNA-seq data (vi) is prepared using GIV (ii), CIF (i) and the scATAC-seq data with (**v**) a region-to-gene matrix. Using the parameters, either the full kinetic model (when RNA velocity is required), or the Beta-Poisson model (when running speed matters) will be used to generate the scRNA-seq data (vii). scMultiSim uses a multiple-step approach that considers both time and space when CCI is enabled (viii). With the simulated true counts (viv), technical noise and batch effects can be added to obtain the observed counts (x).

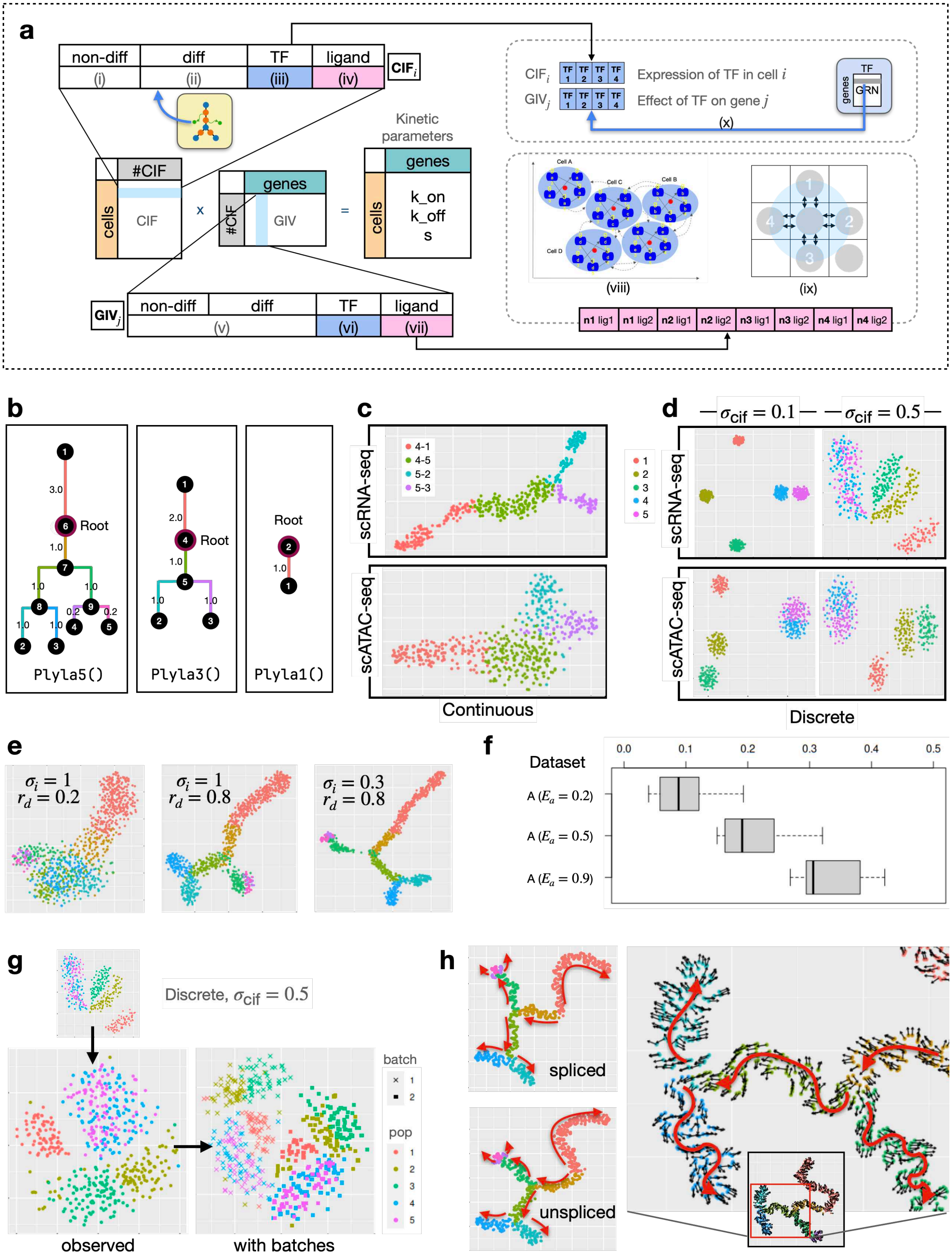

Figure 2. scMultiSim generates multi-modal single cell data from pre-defined cell clustering structure or trajectories. See the next page for descriptions.

Figure 2. scMultiSim generates multi-modal single cell data from pre-defined cell clustering structure or trajectories. **(a)** The CIF and GIV matrix. We multiply the CIF and GIV matrix to get the cell×gene matrix for each kinetic parameter. CIFs and GIVs are divided into segments to encode different biological effects, where each segment encodes a certain type of biological factor. Cellular heterogeneity is modeled in the CIF, and regulation effects are encoded in the corresponding GIV vector. (viii) is the illustration of the cell-cell interactions and in-cell GRN in our model. (ix) is the grid system representing spatial locations of cells. A cell can have at most four neighbors (labeled 1-4) within a certain range (blue circle). The cell at the bottom right corner is not a neighbor of the center cell. (**b**) Three trees are provided by scMultiSim and used to produce the datasets. Phyla1 is a linear trejectory, while Phyla3 and Phyla5 has 3 and 5 leaves, respectively. (**c**) t-SNE visualization of the paired scRNA-seq and scATAC-seq data (without adding technical noise) from the main dataset `MT3a` (continuous populations following tree Phyla3), both having $n_{\text{cell}} = n_{\text{gene}} = 500$. (**d**) t-SNE visualization of the paired scRNA-seq and scATAC-seq data (without adding technical noise) from the main datasets `MD3a` and `MD9a` (discrete populations with five clusters, following tree Phyla5). (**e**) Additional results showing the effect of σ_i and r_d using datasets A. (**f**) Additional results exploring the ATAC effect parameter E_a using datasets A. Averaged Spearman correlation between scATAC-seq and scRNA-seq data for genes affected by one chromatin region, from 144 datasets using various parameters (σ_i, σ_{cif}, r_d, continuous/discrete). All box plots in this article use the standard configuration, i.e., middle lines are medians, boxes represent the 1st and 3rd quartiles, and whiskers are ± 1.5 IQR. (**g**) The observed RNA counts in dataset MD9a with added technical noise and batch effects. (**h**) The spliced true counts, unspliced true counts, and the RNA velocity ground truth from dataset V. The velocity vectors point to the directions of differentiation indicated by red arrows, from the tree root to leaves.

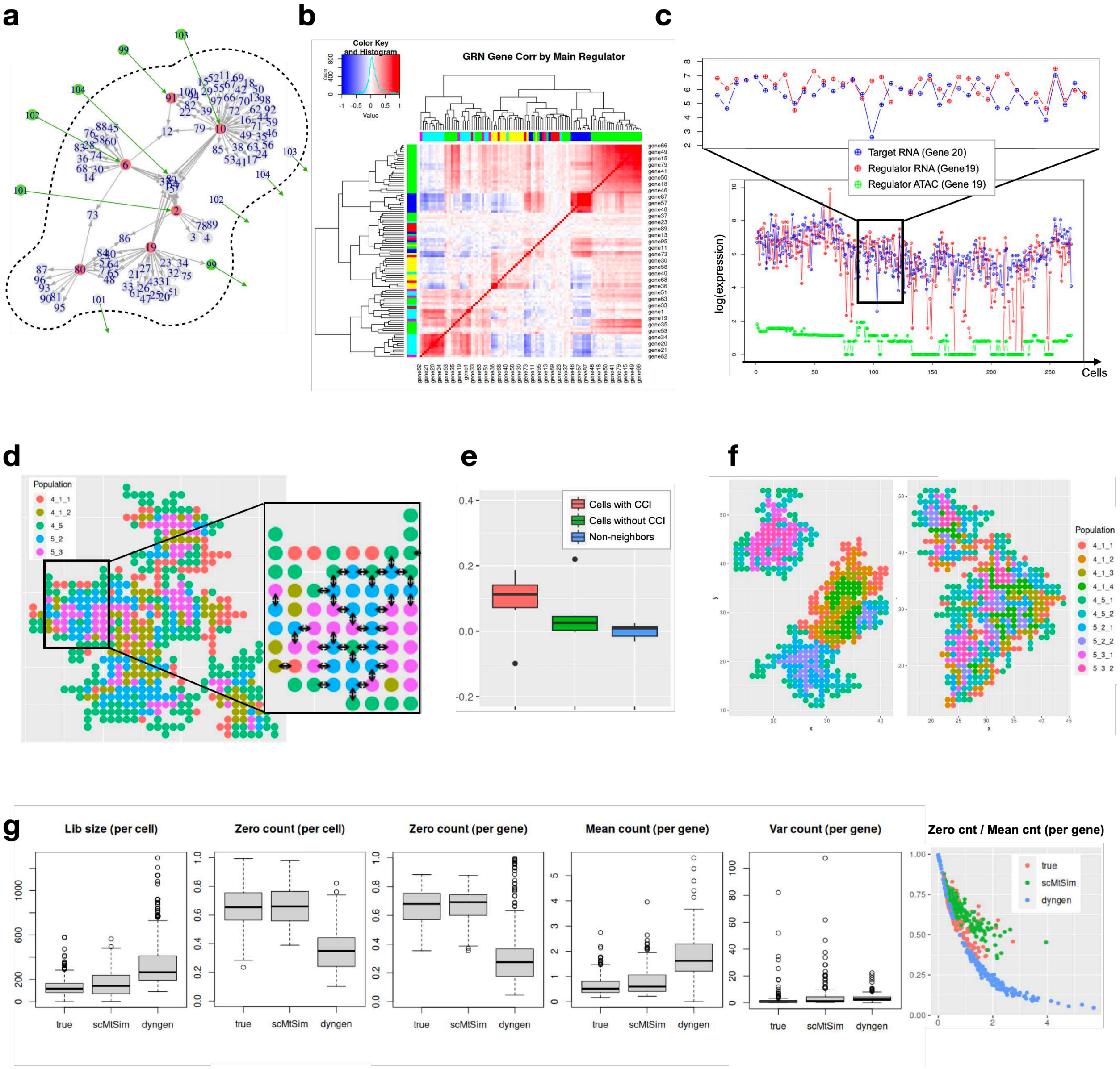

Figure 3. scMultiSim generates realistic single cell gene expression data driven by GRN and cell-cell interaction. See the next page for descriptions.

Figure 3. scMultiSim generates realistic single cell gene expression data driven by GRN and cell-cell interaction. (**a**) The GRN and CCI networks used to generate the main datasets. Red nodes are TF genes and green nodes are ligand genes. Green edges are the added ligand-receptor pairs when simulating cell-cell interactions. (**b-e**) Results from dataset MT3a, which uses Phyla3, 500 genes, 500 cells and $\sigma_{\text{cif}} = 0.1$. (**b**) The gene module correlation heatmap. The color at left or top represents the regulating TF of the gene. Genes regulated by the same TF have higher correlations and tend to be grouped together. (**c**) The log-transformed expression of a specific TF-target gene pair (gene19-gene20) for all cells on one lineage (4-5-3 in Phyla3). Correlation between the TF and target expressions can be observed. We also show the chromatin accessibility level for the TF gene 19, averaged from the two corresponding chromatin regions of the gene. Significant lower expression of gene 19 can be observed when the chromatin is closed. (**d**) The spatial location of cells, where each color represents a cell type. Arrows between two cells indicates that CCI exists between them for a specific ligand-receptor pair (gene101-gene2). By default, most cell-cell interactions occur between different cell types. (**e**) Gene expression correlation between (1) neighboring cells with CCI, (2) neighboring cells with CCI, and (3) non-neighbor cells. Cells with CCI have higher correlations. (**f**) scMultiSim provides options to control the the cell layout. We show the results of 1200 cells using same-type probability $p_n = 1.0$ and 0.8, respectively. When $p_n = 1.0$, same-type cells tend to cluster together, while $p_n = 0.8$ introduces more randomness. (**g**) Comparison between a real dataset and simulated data using multiple statistical measurements (Methods). Parameters were adjusted to match the real distribution as close as possible.

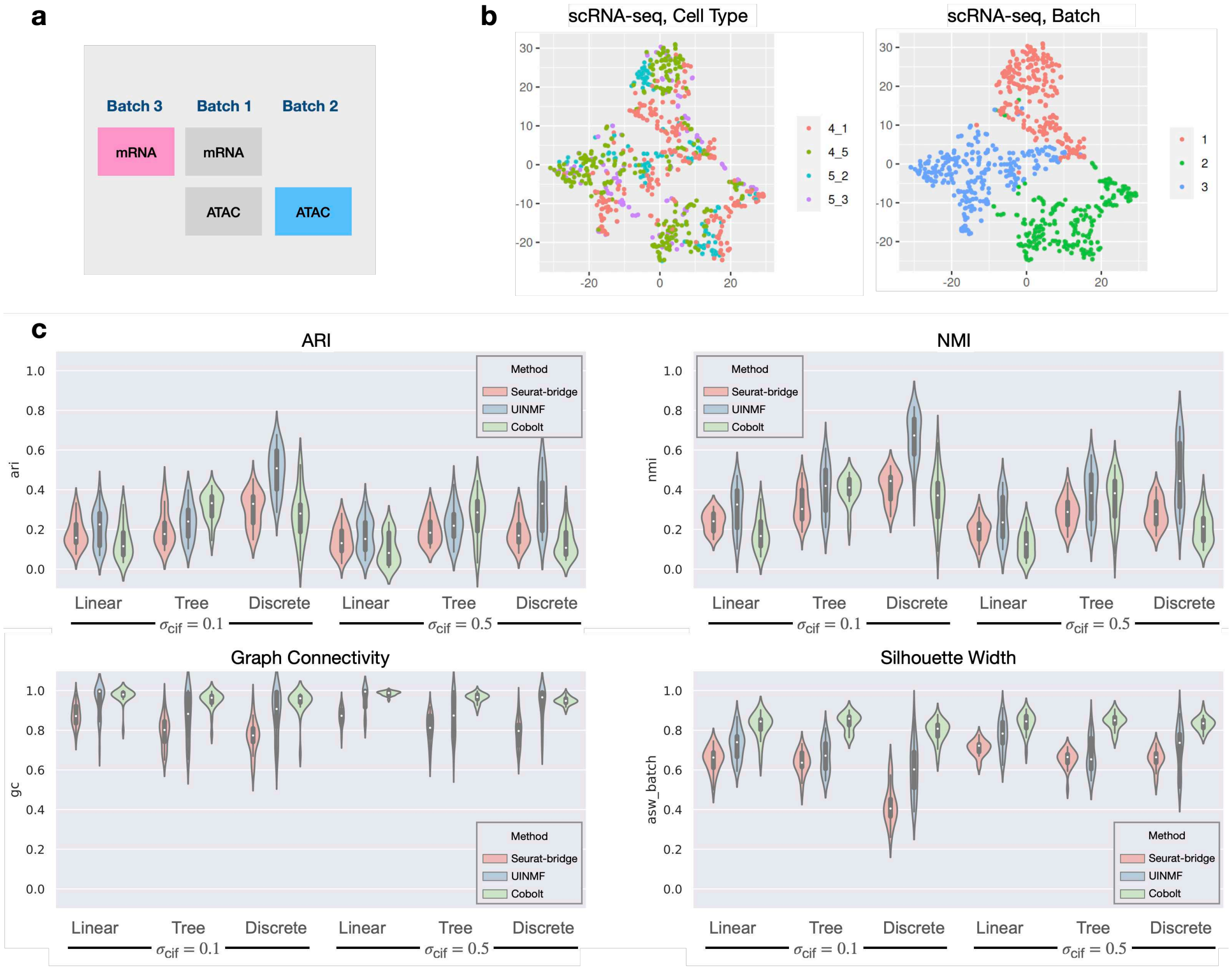

Figure 4. Benchmarking mosaic data integration methods. (**a**) The task illustration of multi-modal data integration. Only cells in batch 1 and 3 (pink and blue matrices) are used for evaluation. (**b**) t-SNE visualization of dataset `MT10a`, cells colored by cell types and batch identities. (**c**) Benchmarking results for mosaic data integration methods, grouped by different trajectory types and σ_{cif}. Metrics used are ARI, NMI (higher = better at preserving cell identities), graph connectivity and average silhouette width of batch (higher = better merging batches).

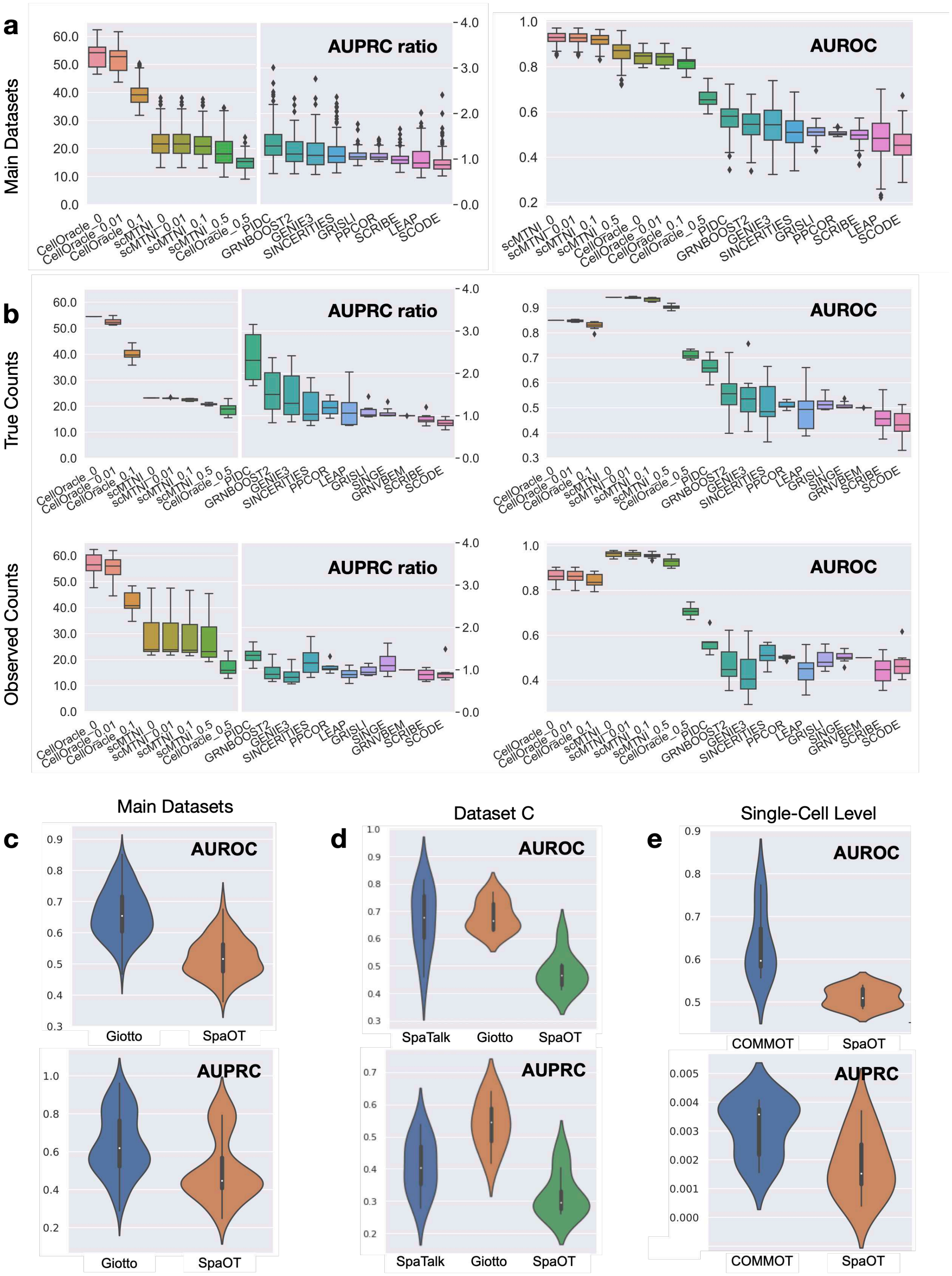

Figure 5. Benchmarking GRN and CCI inference methods.

54

Figure 5. Benchmarking GRN and CCI inference methods. (**a**) Benchmarking GRN inference methods. The box plots show AUPRC ratios (versus a random classifier) and AUROC values. Methods that did not finish in a reasonable time (8h) are excluded. For AUPRC ratio in **a** and **b**, expression-only methods use the Y axis on the right, while multi-modal methods (CellOracle and scMTNI) use a different Y axis on the left due to the huge performance difference. For the two multi-modal methods, we also compare different noise levels (0, 0.01, 0.1, 0.5) added to the gene-peak ground truth matrix (Methods). Direct comparison of all methods can be found in Fig. S6. (**b**) Additional results on benchmarking GRN inference methods using datasets `G` that does not contain CCI effects. We also tested the performance on observed counts with technical noise. (**c**) Benchmarking CCI inference methods on the main datasets. Top: AUROC, bottom: AUPRC. (**d**) Benchmarking CCI inference methods on dataset `C`, with SpaTalk included. Top: AUROC, bottom: AUPRC. (**e**) Benchmarking single-cell-level CCI inference methods on dataset `S`. Top: AUROC, bottom: AUPRC. The AUPRC ratio baseline for this task is 0.0012.

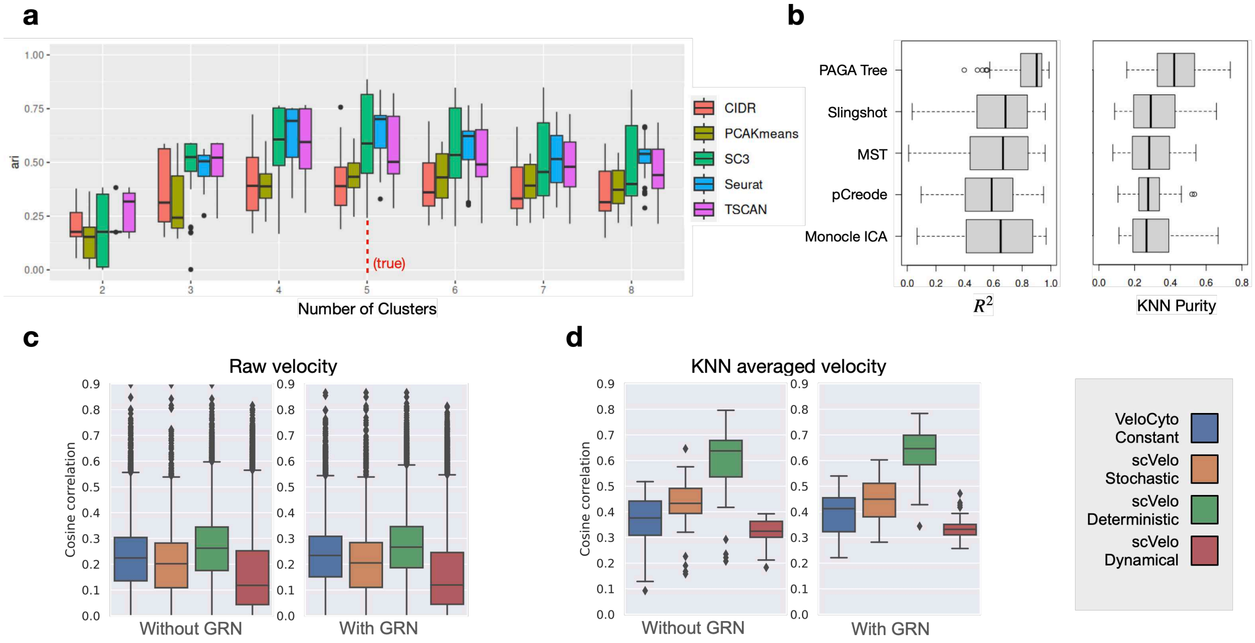

Figure 6. Benchmarking RNA velocity estimation, clustering, and trajectory inference methods. (**a**) Benchmarking clustering methods on dataset `MD` (discrete). Methods are grouped by number of clusters in the result. The vertical red dashed line shows the true number of clusters. A higher ARI indicates better clustering. (**b**) Benchmarking trajectory inference methods on dataset `MT` (continuous tree). Methods are evaluated based on their mean R^2 and kNN purity on each lineage (higher is better). (**c,d**) Benchmarking RNA velocity estimation methods on auxiliary dataset `V`. (**c**) shows the the cosine similarity over raw velocity vectors for each cell. (**d**) shows the the cosine similarity over kNN averaged velocity vectors for each cell.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- scMultiSimsupplementary.pdf