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

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, China

Simulated single-cell data is widely used to aid in designing and benchmarking computational methods due to the scarcity of experimental ground truth. Recently, an increasing number of computational methods have been developed to address various computational problems with single-cell data, including cell clustering, trajectory inference, integration of data from multiple batches and modalities, inference of gene regulatory networks (GRNs) and cell-cell interactions. Simulators that are designed to test a certain computational problem model only the particular factors that affect the output data; whereas modelling as many biological factors as possible into the simulation allows the generated data to have realistic complexity and can be used to benchmark a wider range of computational methods. Here, we present scMultiSim, an *in silico* simulator that generates multi-modality data of single-cells, including gene expression, chromatin accessibility, RNA velocity, and spatial cell locations while accounting for the relationships between modalities. We proposed a unified framework to jointly model biological factors including cell-cell interactions, with-in-cell GRNs and chromatin accessibility, so all their effects simultaneously present in the output data. Users enjoy unprecedented flexibility by having full control of the cell population and the ability to fine-tune each factor's effect on the underlying model. We also provide options to simulate technical variations including batch effects to make the output resemble real data. We verified the simulated biological effects and demonstrated scMultiSim's applications by benchmarking four computational tasks on single-cell multi-omics data: GRN inference, RNA velocity estimation, integration of single-cell datasets from multiple batches and modalities, and analysis of cell-cell interaction using the cell spatial location data. To our knowledge, scMultiSim is the only simulator of single cell data that can perform benchmarking for all these four challenging computational tasks.

Introduction

In recent years, technologies which profile the transcriptome and other modalities (multi-omics) of single cells have brought remarkable advances in our understanding of cellular mechanisms. For example, technologies were developed to profile the chromatin accessibility jointly with the gene expression data [10; 9; 24]; abundance of surface proteins can also be profiled together with the transcriptome [37; 30]; in addition, spatial locations of cells can be measured together with transcriptome profiles using imaging based technologies [34; 15] or sequencing-based [36; 33] technologies. The multi-omics data of single cells allow researchers to study the state of cells in a more comprehensive manner, and more importantly, to explore the relationships between modalities and the causality across hierarchies. Computational methods aiming at integrating multiple modalities of single cell data have been developed [38; 42; 1]. While the current modality integration methods mainly focus on leveraging information from multiple modalities to study cell identities, a small number of them start to study the causal or regulatory relationships cross modalities [47].

The single cell level data also allows researchers to infer gene regulatory networks (GRNs) with large-scale datasets [31]. In addition, the spatial location data allows us to study the cell-cell interactions (CCI) [26; 5; 6]. The current methods which perform GRN inference mostly consider the transcription factors as the only factor which affects the observed gene-expression data. However, the observed gene-expression data is affected by multiple factors, one of which is the chromatin accessibility of the corresponding regions for a target gene. New methods have been developed to use both scRNA-seq and scATAC-seq data to infer GRNs [19; 41; 44]. RNA velocity, on the other hand, can be inferred from the unspliced and spliced mRNA counts to indicate the state of each cell in the near future [22; 2].

Due to the scarcity of experimental data, these studies usually utilize simulators to generate synthetic data resembling real data's statistical features. These simulators have been widely used to aid in designing and benchmarking computational methods [32], and benefiting from the adjustable parameters, they can reveal the impact of a particular biological factor on a method's performance. Several *in silico* simulators have been proposed over the past few years, each taking into account certain biological factors. Earlier simulators like Splatter [43] aim to fit the real scRNA-Seq data to a probability distribution; SymSim [45] is able to simulate multiple scenarios, including discrete and continuous populations and differentially expressed genes. simATAC can simulate scATAC-seq data via a statistical model [28], but it cannot model different types of cell populations. More recent simulators can incorporate the information of a given GRN, like SERGIO [11] and BEELINE [31], or model the spatial cell-cell interaction, like mistyR [40]. Other simulators like scDesign2 [39] focus on generating realistic data by learning distributions from experimental datasets.

While the above aspects are currently studied separately, in biology, chromatin accessibility, GRNs and CCIs are closely related; they together regulate observed gene-expression levels and RNA velocity. Therefore, recent simulators have been trying to incorporate more factors and output more data modalities. Some simple simulation procedures have been used in some papers to simulate both gene expression and chromatin accessibility data [14; 18; 25]. Still, they are oversimplified, and no association between the scATAC-Seq and scRNA-Seq data is modelled. More lately, dyngen [8] can simulate gene expression and RNA velocity affected by within-cell GRNs.

Viewing the current works, we can see each models one or a small subset of the following biological factors: cell population, chromatin accessibility, GRNs, spatial cell locations and cell-cell interactions. They also output one or two of these data modalities: gene expression, RNA velocity, and chromatin accessibility. In contrast to the fact that new computational methods jointly use multiple modalities to uncover more information, to our best knowledge, existing simulators still face challenges incorporating multi-modal data. In this paper, however, we present a unified framework where *all* these biological factors and modalities are modelled and simulated while considering the cross-modality relationships. This simulator, scMultiSim, can thus be used to benchmark computational methods for various contemporary tasks. Fig. 1 shows the input, output and use cases of scMultiSim. Only requiring a cell differential tree and an optional GRN as the input, it simulates discrete or continuous cell populations and outputs the ground truth of all the biological factors. Its applications include but are not limited to clustering, trajectory inference, cell-specific GRN inference, CCI inference using cell location data, RNA velocity estimation, and multi-modal data integration.

In this article, we will first briefly introduce the core concepts and the simulation process of scMultiSim, then demonstrate its capability of simulating various factors simultaneously by statistically validating each of their effects in the data. Finally, we showcase the rich potential applications of scMultiSim by benchmarking computational tools on four different tasks using the simulated data: GRN inference, RNA velocity estimation, multi-modal data integration and CCI inference.

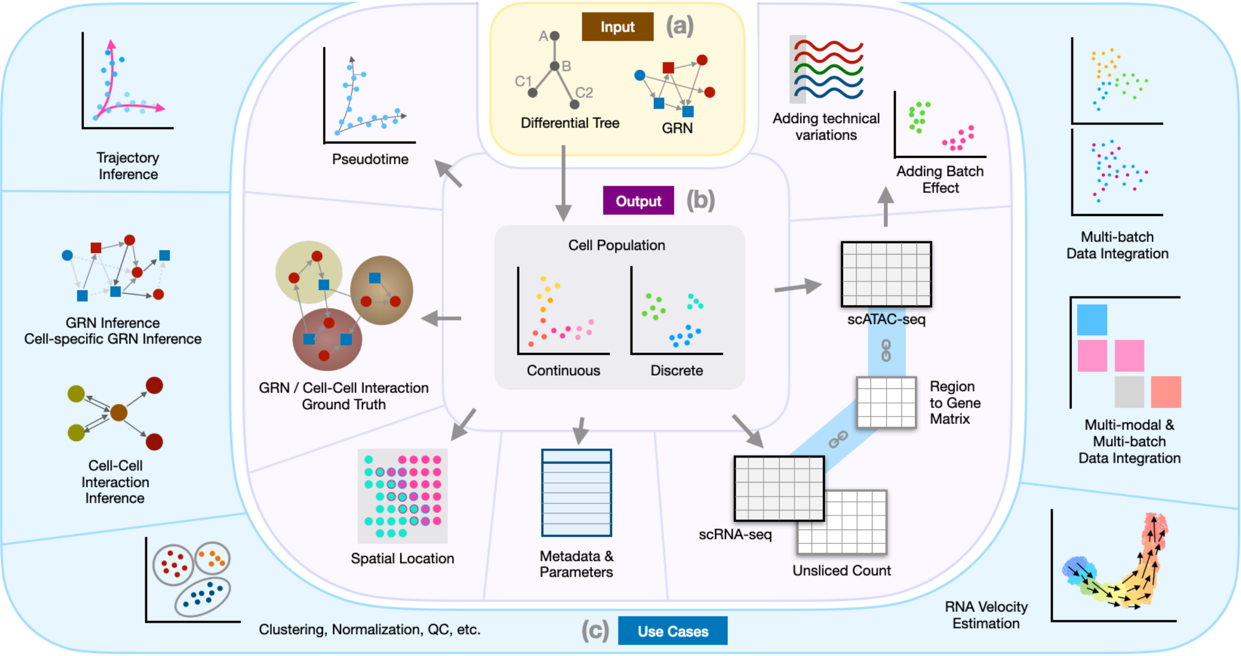

Figure 1. Overview of scMultiSim's input, output, and use cases. (**a**) The minimal required input is a cell differential tree designating the cell population. A user-input GRN is recommended. Users can also control each simulated biological effect using various parameters. (**b**) The output of scMultiSim. (**c**) The major use cases of scMultiSim.

Methods

In general, scMultiSim runs the simulation in two phases (Fig. 3). The first is to generate the true gene expression levels in cells ("true counts"); then, we add technical noise such as library preparation noise and batch effects to get scRNA-seq and scATAC-seq data that is statistically comparable to real data ("observed counts"). In the first phase, we use the widely-accepted kinetic model [29] to simulate gene expression dynamics and generate the RNA velocity. scMultiSim models the cellular heterogeneity and stochasticity of gene regulation effects through a mechanism with two main concepts: Cell Identity Factors and Gene Identity Vectors. They are used to generate the scATAC-seq data and prepare the parameters for the kinetic model. We also expand this mechanism to handle time-varying GRNs and spatial cell-cell interactions.

The kinetic model

scMultiSim extends the idea of SymSim [46] when simulating the gene expression. The kinetic model has four parameters for each gene in each cell, namely k_{on}, k_{off}, s and d. A gene can switch between *on* and *off* states, where k_{on} and k_{off} are 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 rate s and degrade at a rate d. It is common to fix d to be 1 and use the relative values for the other three parameters [27], therefore we need to model the remaining three parameters. The central part of the Method Section will focus on encoding the heterogeneity and complex regulation effects in cells and genes into the kinetic parameters, producing the k_{on}, k_{off} and s parameters for each cell-gene combination, i.e. a cell×gene matrix.

We also provide two modes to generate true counts from the parameters: One is the full kinetic model, where we explicitly let genes undergo several cell cycles with state changes, and the spliced/unspliced RNA counts are calculated. RNA velocity ground truth is also produced in this mode since the RNA synthesize rate is known. The other is the Beta-Poisson model, which is equivalent to the kinetic model's master equation [20]:

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

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

When RNA velocity is unneeded, users can optionally switch to the Beta-Poisson model for faster running time. We

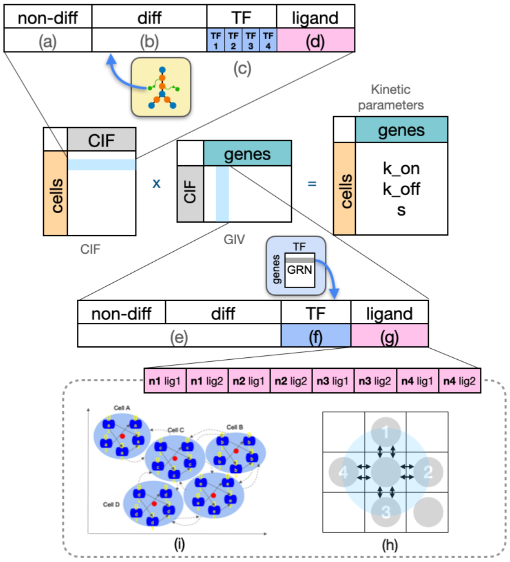

Figure 2. The CIF and GIV matrix. We multiply the CIF and GIV matrix to get the matrix for each kinetic parameter. CIFs and GIVs are divided into segments to encode different biological effects. (**a-d**) Segments of a CIF vector for a cell. Each segment encodes a certain type of biological factor. CIF vectors from all cells form the $n_{\text{gene}} \times n_{\text{cif}}$ CIF matrix. (**e-g**) Segments of a GIV vector for a gene, corresponding to the CIF vector. Combining them, we have the $n_{\text{cif}} \times n_{\text{gene}}$ GIV matrix. The kinetic parameter matrix is obtained by multiplying the CIF and GIV matrix. (**i**) Illustration of the cell-cell interactions and in-cell GRN in our model. (**h**) 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.

introduce an intrinsic noise parameter σ_i that controls the weight of random samples from the Poisson distribution:

$$\text{expr} = \sigma_i \cdot x + (1 - \sigma_i) \cdot (\frac{k_{\mathit{on}}}{k_{\mathit{on}} + k_{\mathit{off}}} \cdot s)$$

The intrinsic noise in the scRNA-seq data originates from the transcription burst and the snapshot nature of scRNA-seq data which is hard to reduce in experiments. This parameter allows users to investigate the effect of intrinsic noise on the performance of the computational methods.

Modeling cellular heterogeneity and various biological effects

Compared to existing simulators, scMultiSim provides unprecedented flexibility to users that (i) encodes various types of biological factors simultaneously, including cell population, chromatin accessibility, GRN and cell-cell interaction, and (ii) produces arbitrary user-defined trajectories in the simulated data while maintaining all the biological effects. In order to model all these factors, we introduce the main concepts of our mechanism: *Cell Identity Factors (CIF)* and *Gene Identity Vectors (GIV)*, which are low-dimension representations of cells and genes (Fig. 2).

The CIF of a cell is a 1D vector representing various biological factors that contributes to the cellular heterogeneity, such as concentrations of proteins and morphological properties. Similar to the EVF concept in SymSim [46], it captures the extrinsic variation that controls the expression pattern (as opposed to the inherent noise in the transcription process). The length of this vector, 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, 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, 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. For example, a higher value in the k_{on} GIV vector $\text{giv}_i^{k_{on}}$ can be interpreted as: higher concentration of the corresponding cell factor $\text{cif}_i^{k_{on}}$ causes a higher activation rate for the gene's promoter. 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.

Encoding various factors in CIF and GIV. A CIF vector can be divided further into four segments (Fig. 2a-d), each represents one type of extrinsic variation. (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 population. These are the biological conditions that are unique to certain cell types. These factors lead to different cell types in the data. For a heterogeneous cell population, cells have different development statuses and types. Values for diff-CIFs are used to represent these cell differential factors, which are generated based on the user-input cell differential tree. (iii) CIFs corresponding to Transcription Factors (**tf-CIF**) control the GRN effects. This segment models how a TF can affect expression of genes in the cell. Its length equals to the number of TFs. (iv) CIFs corresponding to ligands (**lig-CIF**) from neighboring cells control the effect of CCI. If CCI simulation is enabled, this segment models effects from ligands of a neighbor cell. More details are included in Supp. A. We

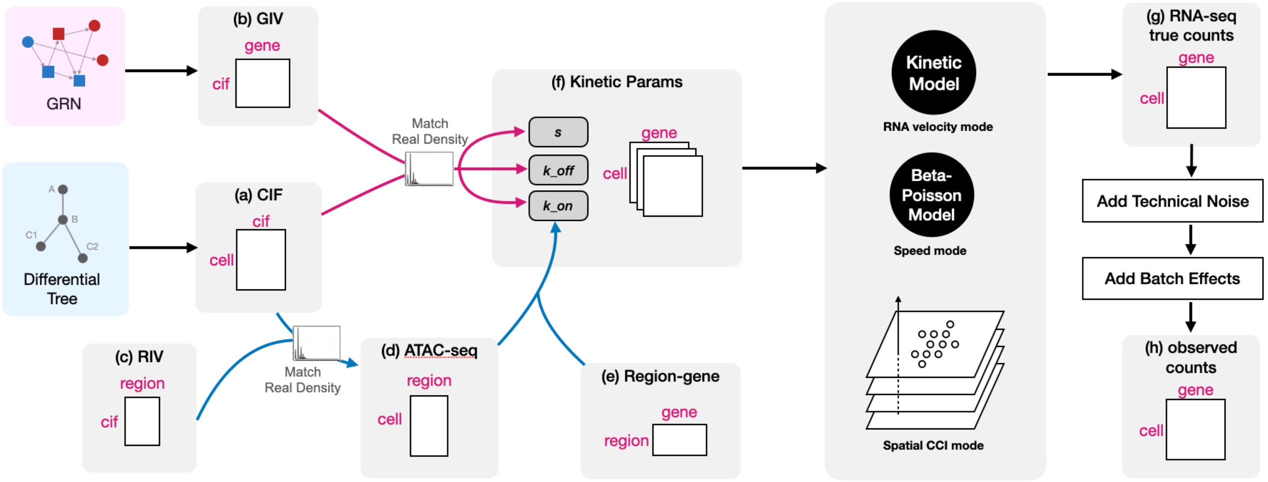

Figure 3. The overall structure of scMultiSim.

also generate the corresponding GIV matrices carefully considering the nature of the kinetic model (see following sections, and Supp. B).

diff-CIF generates user-controlled trajectories. The minimal user input of scMultiSim is the cell differential tree, which controls the cell types (for discrete population) or trajectories (for continuous population) in the output. The tree is modeled by the diff-CIF vectors (Fig. 2b): starting from the root of the tree, a Gaussian random walk along the tree (Supp. A) 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 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.

tf-CIF and GIV encode the GRN effects. The TF part of the GIV (Fig. 2f) is a $n_{\text{gene}} \times n_{\text{TF}}$ matrix. Naturally, we put the GRN effect matrix here for 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 expression. The corresponding tf-CIF (Fig. 2c) is sampled randomly for the cell at the tree root; for a downstream cell in the tree, its tf-CIF is a scaled version of the TF expression of the previous cell. In this way, we can simulate the inheritance relationship between cells along the differential tree.

lig-CIF and GIV encode cell-cell interactions. The CCI is modeled using a similar method as the GRN. However, now the cell's receptor is affected by ligands of multiple neighbors (Fig. 2i). We model the spatial location of cells using a grid (Fig. 2h), where a cell can have at most four neighbors with CCI (within the blue circle's range). Therefore, the ligand CIF and GIV (Fig. 2 d/g) are of length $4 \cdot n_{\text{LR}}$, where n_{LR} is the number of ligand-receptor pairs. The lig-GIV vector contains the CCI strength values, for example, the "n1lig2" entry is how strong the ligand 2 from the neighbor at position 1 can affect the receptor 2 of this 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.

The simulation process

Fig. 3 shows an overview of the simulation process. First, scMultiSim generates the scATAC-seq data (Fig. 3d, more details in Supp. C). Similar to GIV, we use a randomly sampled *Region Identity Vector (RIV)* matrix to model the chromatin regions. Following the same mechanism, we get the raw scATAC matrix by multiplying the CIF and RIV matrix, which represents the open chromatin regions in each cell. Next, the scATAC-seq data is obtained by scaling the raw matrix to match a real distribution learned from real data (Supp. F). This is an important step to capture the intrinsic variation of the chromatin accessibility pattern, which we will also apply for the kinetic parameters when generating gene expressions.

Afterwards, CIF and GIV matrices are used to generate the three kinetic parameter matrices. To obtain coupled scATAC-seq and scRNA-seq data, the k_{on} parameter is determined jointly by the scATAC-seq data and the result from GIV, because chromatin accessibility controls the activated status of genes. Specifically, the matrix from GIV is used as a reference to fill the zero entries in the sparse matrix from scATAC-seq data so that the entries can be differentiated (Supp. D). A region-to-gene matrix (Fig. 3e) is also generated to represent the mapping between chromatin regions and genes, where a gene can be regulated by 1-3 consecutive regions. When obtaining the k_{on}, k_{off} and s parameters, we also scale the raw values from matrix multiplication to match the distribution sampled from real data. With the parameters, the true counts can thus be simulated using the Kinetic model.

If the full kinetic model is used (as opposed to the Beta-Poisson model), cells undergo several cycles before the spliced and unspliced counts are outputted. In each cycle, the cell may switch between on/off state with probabilities determined by the parameters (Supp. G). The RNA velocity ground truth is calculated using the real splicing and degradation rate of each gene. Otherwise, if the Beta-Poisson model was used, the true counts are simulated according to the equations. No RNA velocity and unspliced count data is outputted in this mode.

Simulating cell-cell interaction. If spatial cell-cell interaction is enabled, scMultiSim instead uses a multiple-step approach to replace the original kinetic model (fig. 3, Spatial CCI mode). This mode models both time and space – in Fig. 3 “Spatial CCI mode”, each layer corresponds to a time point. Cells are placed in a grid as described before, and one cell is added to the grid at each step, representing a newborn cell. The new cell will always be in the initial state at the root of the differential tree. Strategies are designed to ensure cells have a similar number of neighbors in the grid, and similar cell types are closely located (Supp. E). At each step, an existing cell moves forwards along a random path in the cell differential tree, representing the cell development. The CCI and GRN effects are simulated for all cells using the Beta-Poisson model. We output the final step as the result, which contains the accumulated CCI effects during the cells’ development process.

Multiple techniques are used to maintain the population structure (clusters or trajectories) in the final output. First, 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. Next, we let the same cell (at the same location) having 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 are inherited from the previous one to make sure other factors stay the same.

The minimal user input for simulating CCI is a list of ligand-receptor pairs. In reality, CCI only happen between some specific cell types. Therefore, scMultiSim first divides cell into types based on their development stage on the tree, then randomly generate the CCI ground truth by enabling a random set of ligand-receptor pairs for every two cell types. It also makes sure that very few interactions occur between two cell of the same type. User can choose to input the ground truth to further control the behavior.

Incorporating dynamic GRN. The GRN can be set to a time-varying mode. 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.

Technical variations and batch effects

scMultiSim can simulate the library preparation procedure for both true scRNA and scATAC counts to generate observed counts. The workflow follows SymSim’s approach [46]: we simulate multiple rounds of mRNA capture and PCR amplification, then sequencing and profiling with UMI or non-UMI protocols. Batch effects are added by first dividing the cells into batches, then adding gene-specific and batch-specific Gaussian noise. The strength of the batch effects can also be controlled via parameters.

Results

As a fundamental requirement of simulators, we first show that scMultiSim’s output can statistically resemble real data. Therefore, we prepared a reference real dataset (Supp. H), and compared it with scMultiSim and dyngen [8] using various metrics shown in Fig. 6. Fig. 6a shows that both scMultiSim and dyngen are able to simulate data with proper library size. For the proportion of zero counts (Fig. b-c), this experimental dataset shows low variance per cell but high variance per gene, which brings challenges for simulators to match with them; nevertheless, scMultiSim performs better in simulating the large variance of zero counts per gene. From Fig. 6c-e, we observe that scMultiSim successfully simulates the high variance of counts per gene, reflecting the phenomenon where a small proportion of genes are highly expressed in experimental data. Dyngen was not able to capture this distribution. There is also usually a negative correlation between zero counts and mean counts in real data (Fig. 6f), and scMultiSim was able to simulate this relationship.

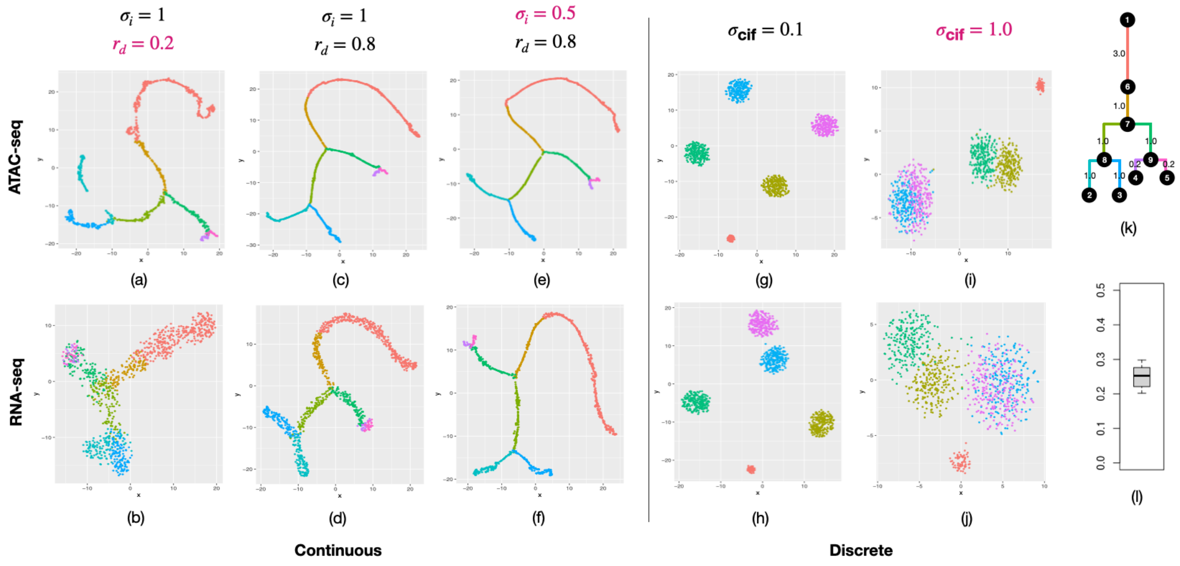

Figure 4. scMultiSim generates scRNA-seq and scATAC-seq from pre-defined cell clustering structure or trajectories. (**a**)-(**j**) tSNE visualization of the scRNA and scATAC data with different parameter sets, where the first row is scATAC-seq data, and the second row is scRNA-seq data. (**k**) The differential tree used. (**l**) Averaged Spearman correlation between scATAC-seq and scRNA-seq data for genes affected by one chromatin region. The box plot contains 20 datasets using various parameters (σ_i, σ_{cif}, r_d, continuous/discrete).

scMultiSim generates scRNA-seq and scATAC-seq from pre-defined cell clustering structure or trajectories

A major advantage of scMultiSim is its ability to generate coupled scRNA-seq and scATAC-seq data from user designated clustering or trajectories. It also provides copious parameters to fine-tune the shape in the cell population. We mainly experiment on these three parameters: intrinsic noise σ_i, CIF sigma σ_{cif}, and diff-to-non-diff CIF ratio r_d, as shown in Fig. 4. From the differential tree (Fig. 4k), scMultiSim generates both continuous (a-f) and discrete (g-j) cell population. All datasets were simulated using 1000 cells, 500 genes, and number of CIFs $n_{\text{cif}} = 50$. CIF sigma $\sigma_{\text{cif}} = 0.1$ is used if not specified in the figure.

For continuous populations, trajectories corresponding to the input tree are clearly visible using the default parameters (c-d). With a smaller diff-to-non-diff CIF ratio r_d (a-b), the differential CIF (where the tree is encoded) has less control on the expression. Therefore, the trajectory is vague and more randomness is introduced. With a smaller intrinsic noise σ_i (e-f), a fraction of the expression value is directly calculated from kinetic parameters without sampling from the Poisson model. As a result, the trajectory is more prominent and crisp. These patterns are much cleaner than real data because real data always has intrinsic noise. Together with other metadata ground truth such as pseudotime and cell types, scMultiSim can be used to benchmark tools focusing on trajectory data.

scMultiSim is also capable of generating discrete population using the cell types at the tips of the tree (five cell types as shown in g-j). The parameter σ_{cif} controls the standard deviation of the CIF, therefore with a smaller σ_{cif} (i-j), the clusters are more disorganized. With the true cluster labels, scMultiSim can be used to benchmark clustering methods.

Correlation between scATAC-seq and scRNA-seq data. In order to validate the connection between the scATAC-seq and scRNA-seq data, we calculate the mean spearman correlation between them for each gene. While we can observe correlation directly between the scATAC and scRNA data, since one gene can be controlled by 1-3 chromatin regions in scMultiSim, the effect is complicated and the correlation is less prominent. Stronger correlation can be revealed if we only consider genes affected by one single region. In Fig. 4l, an averaged 0.2-0.3 correlation is observed across different parameter settings, indicating the connection between the two modalities.

scMultiSim generates single cell gene expression data driven by GRN and cell-cell interaction

The strength of scMultiSim also resides in its ability to incorporate the effect of GRN and CCI while preserving the trajectory structures. We ran experiment to validate that the GRN and CCI effects co-exist in the simulated expression data. We use the 100-gene GRN from [12] as the ground truth, which is visualized in Fig. 5b. We also enable CCI simulation at the same time by adding ligand-receptor pairs from gene 101-104 to gene 2, 6, 10 (TFs), and 8 (non-TF) in the GRN. A simple bifurcating tree is used as the trajectory and totally 500 cells and 200 genes is simulated. The intrinsic noise is set to 0.5.

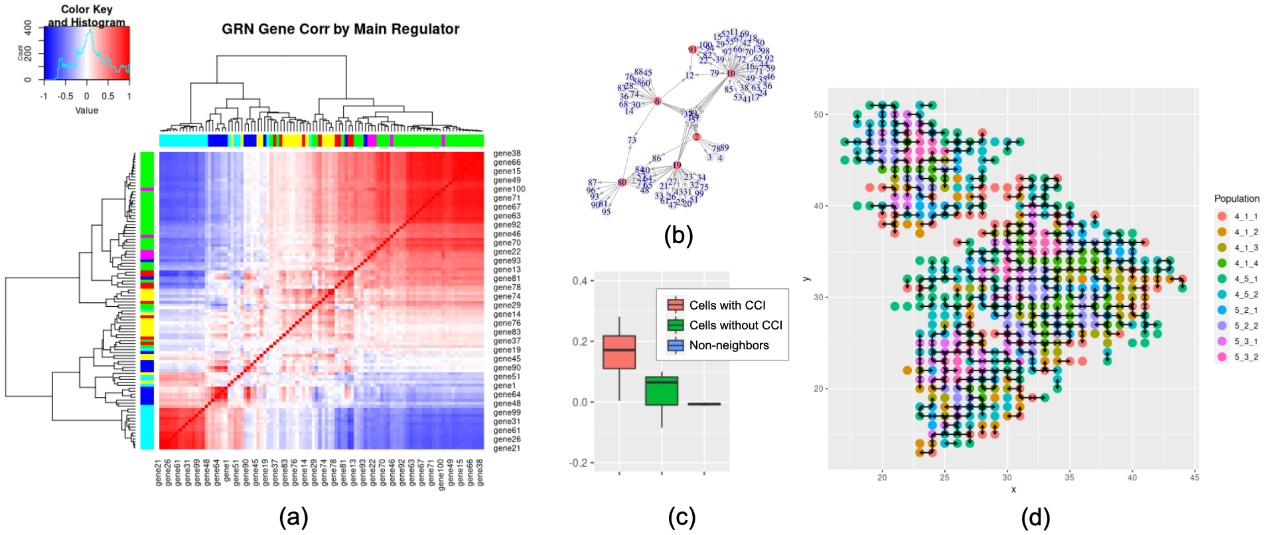

Figure 5. scMultiSim generates single cell gene expression data driven by GRN and cell-cell interaction. (**a**) The gene module correlation heatmap. The color at left or top represents the regulating TF of the gene. Genes regulated by the same TF have higher correlations and tend to be grouped together. (**b**) The GRN used. (**c**) Gene expression correlation between (1) neighboring cells with CCI, (2) neighboring cells with CCI, and (3) non-neighbor cells. (**d**) The outputted spatial location of cells. Each color represents a cell type, An arrow between two cells indicates that CCI exists between them for the specific ligand-receptor pair.

GRN guided expression data. We illustrate the gene regulation effects using a gene module correlation heatmap as shown in Fig. 5a. The clustered heatmap is generated using correlations between each pair of regulated genes. Each color on the top or left side represents a TF in the GRN. The figure clearly shows that gene modules regulated by the same TF (genes with the same color) tend to be grouped together, while having higher correlations with each other. Thus, the heatmap suggests that the effect of GRN presents in the expression data.

Cell spatial locations. scMultiSim provides convenient helper methods to visualize the cell spatial locations as in Fig. 5d. For each ligand-receptor pair, arrows can be displayed between cells to show the direction of cell-cell interactions. Multiple strategies are designed to place the cells in the grid; the default strategy used here is that a newborn cell has a higher probability of staying with a cell of the same type. Therefore, small cell clusters can be observed in the final result, but overall different cell types still mix well to enable more interactions between cell types. In real data, how likely cells from the same cell type locate together depends on the tissue type, and scMultiSim provides a probability parameter to tune this pattern.

Correlations between cells with CCI. scMultiSim generates CCIs between single cells as well as between cell types. We validate the simulated CCI effects by comparing the correlations between (i) neighboring cells with CCI, (ii) neighboring cells without CCI, and (iii) non-neighbor cells. When calculating correlation with non-neighbors, four non-adjacent cells are randomly sampled for the current cell to keep the data scale comparable. Cell pairs of the same type are ignored while calculating the correlations because they tend to have similar expressions. Fig. 5c shows that gene expressions of neighboring cells with CCI have an average correlation of 0.1, while cells without CCI have approximately zero correlation, which is expected. We noticed that neighboring cells without CCI still have slightly higher correlation compared to non-neighbor cells; and we speculate that this is because the cells are evolving to new cell types over time, so the CCI effect involved in an earlier cell type may remain in the final step.

scMultiSim generates RNA velocity data

If RNA velocity simulation is enabled, the kinetic model outputs the velocity ground truth using the RNA splicing and degradation rates. Fig. 6 g-h show both the raw RNA velocity and the velocity averaged by KNN, which can be used to benchmark RNA velocity estimation methods.

Benchmarking GRN inference methods

We benchmarked 11 GRN inference methods which were compared in a previous benchmarking paper [32]. The same 100-gene GRN from [12] is used to simulate 8 datasets with 1000 cells using a linear trajectory. Using the predicted networks, we calculate the AUROC (area under receiver operating characteristic curve) as well as the

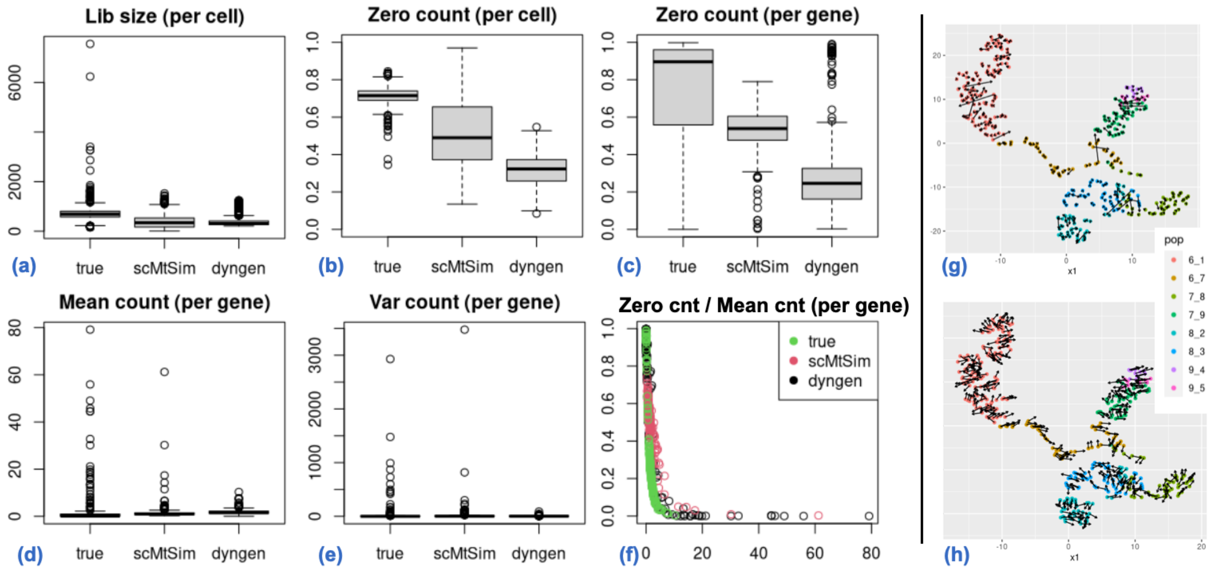

Figure 6. scMultiSim generates realistic data and RNA velocity data. (**a-f**) Comparison of simulated and real data's distributions. (**g-h**) RNA Velocity data. Top: Raw velocity values; bottom: KNN averaged normalized values. Visualized using tSNE.

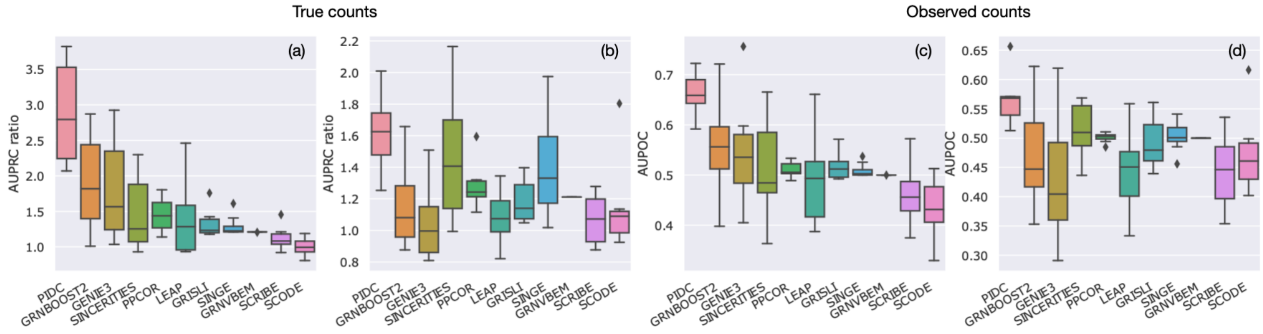

Figure 7. Benchmarking GRN inference methods. The GRN has 130 edges and 6 TFs. There are 110 genes and 1000 cells in total, and a linear trajectory is used. 8 datasets were generated. The first row is AUPRC ratios (versus a random classifier), and the second row is AUROC values. (**a**)-(**b**) The results using true counts. (**c**)-(**d**) The results using observed counts with technical noise.

AUPRC (area under precision-recall curve) ratio, which is the AUPRC divided by the baseline value (in our case, network density) [32]. We ran the benchmark using true counts and observed counts respectively, and the result is shown in Fig. 7. We observed that PIDC has the best overall performance, especially on true counts. Other methods like GENIE3 and GRNBOOST2 also have noteworthy precision. We then examined the effect of technical noise on the performance of GRN inference methods. On observed counts, PIDC continues to have the highest AUPRC and AUROC values, showing that its performance is more resistant to technical noises. When it comes to observed counts, SINCERITIES, PPCOR and SINGE perform well and beats GENIE3 and GRNBOOST2.

Notably, the ordering of the methods tested using true counts is generally consistent with the ordering reported in [32] even though a different groundtruth GRN was used. Nevertheless, the absolute AUPRC values of all methods are still far from satisfying, indicating that GRN inference is still a challenging problem.

Benchmarking RNA velocity estimation methods

We demonstrate scMultiSim's ability of benchmarking RNA velocity estimation methods by running scVelo [3] and VeloCyto [4] on the simulated data. 72 datasets were generated for combinations of {500, 750, 1000} cells and {100, 200, 500} genes. The metric used is the cosine similarity between the estimated velocity and the ground truth after averaged by KNN (Supp. I). From the result shown in Fig. 8, scVelo's deterministic velocity estimation has the best performance on the datasets. The inclusion of GRN results in slightly higher accuracy, which suggests that additional data heterogeneity helps infer RNA velocity. The average cosine of 0.63 is high, showing that methods can accurately estimate the overall RNA velocity. However, if we calculate the cosine similarity directly on each cell

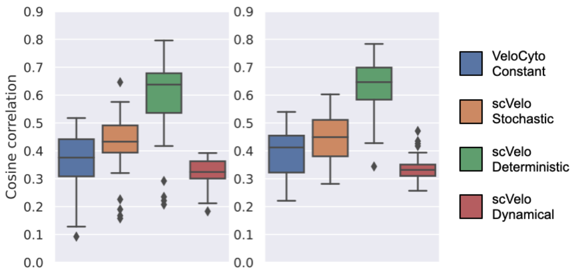

Figure 8. Benchmarking RNA velocity estimation methods. Left: without GRN; right: with GRN. Data was simulated using {500, 750, 1000} cells and {100, 200, 500} genes. 8 datasets were generated for each configuration.

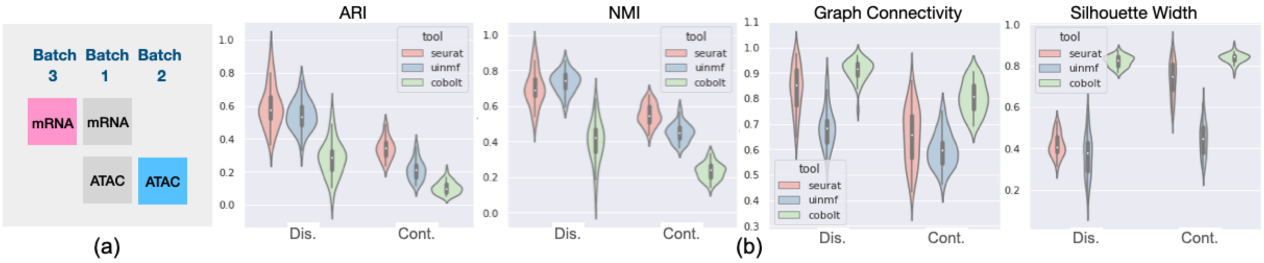

Figure 9. Benchmarking multi-modal data integration methods. (**a**) The task definition. We simulate scRNA and scATAC data with three batches, using the combinations of {1500, 2250, 3000} cells and {100, 200, 500} genes. Four datasets are generated for each configuration. Only cells in batch 1 and 3 (pink and blue matrices) are used for evaluation. (**b**) Integration results visualized using tSNE on a dataset simulated with 3000 cells and 200 genes. (**c**) Metrics for the methods: ARI, NMI (higher = better at preserving cell identities), graph connectivity and average silhouette width of batch (higher = better merging batches).

without KNN normalization, the correlation decreased to about 0.2 (Supp. J, Fig. S4). The correlation values are in line with dyngen [8], which also shows a significant performance difference between averaged (average velocity at trajectory waypoints weighted by a gaussian kernel) and individual RNA velocity vectors, indicating that it is still challenging to infer the velocity for each gene individually.

Benchmarking multi-modal data integration methods

Multi-modal data integration is becoming more popular as they enable downstream analysis utilizing information from multi-omics data and different batches. We benchmarked three new multi-modal integration methods: Seurat bridge integration [17], UINMF [21] and Cobolt [16]. Four datasets were simulated for each possible combination of {1500, 2250, 3000} cells and {100, 200, 500} genes. The simulated data are divided into three batches, then the scRNA data from batch 2 and scATAC data from batch 3 are dropped intentionally to mimic a real scenario where some modalities are lacking (Fig. 9a). Since it is difficult to obtain the latent embedding from Seurat for the "bridge" data (batch 1), only the two matrices coming from batch 2 and 3 (colored in Fig. 9a) were used for evaluation. We use Adjusted Rand Index (ARI) and Normalized mutual information (NMI) as the metrics for cluster identity preservation. Metrics for batch correlation are Graph connectivity and average silhouette width (ASW) of batch (Supp. K). These metrics were used in a recent paper benchmarking single cell data integration methods [23].

The result is shown in fig. 9b. We observe that Seurat bridge integration has the best performance on both continuous and discrete datasets in integrating cell types across modalities and batches. UINMF also has comparable ARI and NMI values with Seurat for discrete populations, but its performance is inferior on continuous datasets. Cobolt is the least satisfactory one among the three. When looking at the capability of merging different batches, Cobolt ranks at the top, while UINMF has the worst performance producing well-mixed batches. The results also suggest that continuous data is more challenging for the integration methods due to the unclear cluster boundaries.

We also visualized the integration results in Fig. S5, which helped us to understand the characteristic of each method's behavior. We noticed that while Seurat has lower graph connectivity and ASW scores, different batches are located closely (but do not overlap) in the visualized latent space. This means that Seurat may be able to identify cells from each batches, but chooses to separate the reference and query data in the latent space. Further inspection of the predicted cell types outputted by Seurat shows very accurate results, indicating that the latent embedding alone cannot represent Seurat's batch merging ability.

Benchmarking CCI inference methods

We benchmarked three CCI inference methods based on spatial cell location data, namely Giotto [13], SpaOTsc [7] and SpaTalk [35]. SpaTalk needs a mininum of 3 genes from the receptor to a downstream activated TF, therefore

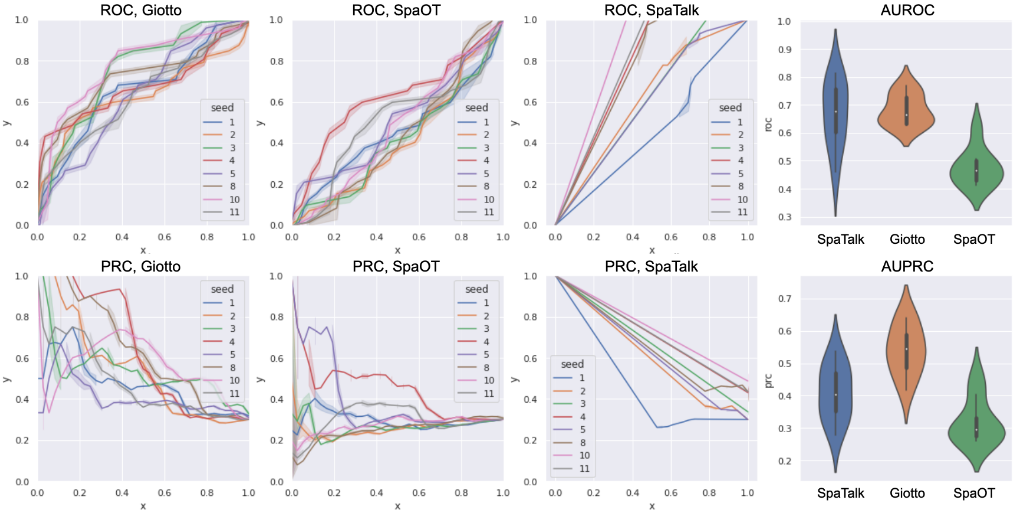

Figure 10. Benchmarking CCI inference methods. First row: ROC curves of Giotto, SpaOTsc, and SpaTalk; AUROC values of the three methods. Second row: PRC curves of Giotto, SpaOTsc, and SpaTalk; AUPRC values of the three methods.

an artificial GRN with long pathways is used to satisfy such requirement. Totally 8 datasets were generated with 500 cells and 160 genes using a linear trajectory. The result is shown in fig. 10. When calculating the PRC and ROC curves, we used Giotto's significance score and the Bonferroni corrected p-values from SpaTalk. Overall, Giotto gives excellent performance by having an average AUROC of 0.68 and AUPRC of 0.54 (baseline is 0.3). SpaTalk has too many identical p-values in the output, making the ROC and PRC curves unrealistic. We speculate that it is due to the artificial GRN still being too simple, indicating that SpaTalk is only designed for large-scale experimental data. Nevertheless, it has noteworthy performance in terms of AUROC and AUPRC values but is less accurate and stable than Giotto. The benchmarking results show that Giotto could be the versatile yet robust choice for CCI inference.

Discussion

We presented scMultiSim, a simulator of single cell multi-omics data which is able to simulate biological factors including cell population, chromatin accessibility, RNA velocity, GRN and spatial cell-cell interactions. We verified presence of the simulated factors in the output data and the relationship between modalities, as well as demonstrated its applications through benchmarks on various computational problems. Furthermore, by obtaining consistent benchmarking results with previous works like BEELINE [32] and dyngen [8], the simulated biological effects are validated to be practical and ready for real-world use.

Compared to existing simulators that focus on simulating one biological factor, data simulated by scMultiSim involves more biological factors, therefore has additional complexity similar to real data. Researchers using scMultiSim can therefore better estimate their methods' real-world performance on noisy experimental data. Furthermore, with the coupled data modalities in the output, researchers can benchmark their computational methods utilizing multiple modalities, which was previously impossible.

scMultisim's modal is easy to extend to include more biological factors and modalities. Its flexibility and many functionalities are enabled by its mechanistic nature, which considers biological mechanisms as much as possible. Moreover, the framework we used to model chromatin regions (RIV) and genes (GIV) can also be expanded to other data modalities, like the protein abundance data. We have shown that our CIF/GIV model is versatile enough to not only mathematically represent the effects, but also incorporate various biological mechanisms.

We underline that scMultiSim's major advantage is encoding various factors using 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, which meets the needs of designing and benchmarking new methods on tasks involving multi-omics data. We believe that scMultiSim could be a powerful tool fostering new computational methods for single-cell multi-omics data. With more subsequent benchmarks, it can also help researchers choose their proper tool based on what kind of data they have.

Bibliography

1. 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.
2. 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.
3. V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38(12):1408–1414, 2020.
4. V. Bergen, M. Lange, S. Peidli, F. A. Wolf, and F. J. Theis. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nature Biotechnology*, 38(12):1408–1414, 2020.
5. 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.
6. Z. Cang and Q. Nie. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. 11(1), Apr. 2020.
7. Z. Cang and Q. Nie. Inferring spatial and signaling relationships between cells from single cell transcriptomic data. *Nature Communications*, 11, 2020.
8. 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.
9. 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.
10. 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.
11. P. Dibaeinia and S. Sinha. SERGIO: A Single-Cell expression simulator guided by gene regulatory networks. *Cell Syst*, Aug. 2020.
12. P. Dibaeinia and S. Sinha. SERGIO: A single-cell expression simulator guided by gene regulatory networks. 11(3):252–271.e11, Sept. 2020.
13. 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.
14. Z. Duren, X. Chen, M. Zamanighomi, W. Zeng, A. T. Satpathy, H. Y. Chang, Y. Wang, and W. H. Wong. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. *Proc. Natl. Acad. Sci. U. S. A.*, 115(30):7723–7728, July 2018.
15. 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.
16. B. Gong, Y. Zhou, and E. Purdom. Cobolt: integrative analysis of multimodal single-cell sequencing data. *Genome Biology*, 22(1):351, 2021.
17. Y. Hao, T. Stuart, M. 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. *bioRxiv*, 2022.
18. S. Jin, L. Zhang, and Q. Nie. scAI: an unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. *Genome Biol.*, 21(1):25, Feb. 2020.
19. K. Kamimoto, C. M. Hoffmann, and S. A. Morris. CellOracle: Dissecting cell identity via network inference and in silico gene perturbation. Apr. 2020.
20. J. Kim and J. C. Marioni. Inferring the kinetics of stochastic gene expression from single-cell RNA-sequencing data. 14(1):R7, 2013.
21. 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.
22. 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.
23. 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.
24. 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.
25. C. Martínez-Mira, A. Conesa, and S. Tarazona. MOSim: Multi-Omics simulation in R. Sept. 2018.
26. U. Mayr, D. Serra, and P. Liberali. Exploring single cells in space and time during tissue development, homeostasis and regeneration. *Development*, 146(12), June 2019.
27. B. Munsky, G. Neuert, and A. van Oudenaarden. Using gene expression noise to understand gene regulation. 336(6078):183–187, Apr. 2012.
28. Z. Navidi, L. Zhang, and B. Wang. simATAC: a single-cell ATAC-seq simulation framework. *Genome Biol.*, 22(1):74, Mar. 2021.
29. J. Peccoud and B. Ycart. Markovian modeling of gene-product synthesis. 48(2):222–234, Oct. 1995.
30. V. M. Peterson, K. X. Zhang, N. Kumar, J. Wong, L. Li, D. C. Wilson, R. Moore, T. K. McClanahan, S. Sadekova, and J. A. Klappenbach. Multiplexed quantification of proteins and transcripts in single cells. *Nat. Biotechnol.*, 35(10):936–939, Oct. 2017.
31. 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.
32. 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. *Nature Methods*, 17(2):147–154, 2020.
33. 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.
34. 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.
35. 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.
36. 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.
37. M. Stoeckius, C. Hafemeister, W. Stephenson, B. Houck-Loomis, P. K. Chattopadhyay, H. Swerdlow, R. Satija, and P. Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nat. Methods*, 14(9):865–868, Sept. 2017.
38. 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.
39. 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.
40. 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.
41. 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 multi-omics. Sept. 2022.
42. 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.
43. L. Zappia, B. Phipson, and A. Oshlack. Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, 18(1):174, 2017.
44. S. Zhang, S. Pyne, S. Pietrzak, A. F. Siahpirani, R. Sridharan, and S. Roy. Inference of cell type-specific gene regulatory networks on cell lineages from single cell omic datasets. July 2022.
45. X. Zhang, C. Xu, and N. Yosef. Simulating multiple faceted variability in single cell RNA sequencing. *Nat. Commun.*, 10(1):2611, June 2019.
46. X. Zhang, C. Xu, and N. Yosef. Simulating multiple faceted variability in single cell RNA sequencing. *Nature Communications*, 10(1), 2019.
47. 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.