Zhang *et al. Genome Biology* (2022) 23:139
https://doi.org/10.1186/s13059-022-02706-x

Genome Biology

METHOD Open Access

scDART: integrating unmatched scRNA-seq and scATAC-seq data and learning cross-modality relationship simultaneously

Ziqi Zhang[1], Chengkai Yang[2] and Xiuwei Zhang[1*]

*Correspondence:
xiuwei.zhang@gatech.edu
[1] School of Computational Science and Engineering, Georgia Institute of Technology, 30308 Atlanta, GA USA
[2] Department of Electrical Engineering and Information Systems, Graduate School of Engineering, The University of Tokyo, Tokyo, Japan

Abstract

It is a challenging task to integrate scRNA-seq and scATAC-seq data obtained from different batches. Existing methods tend to use a pre-defined gene activity matrix to convert the scATAC-seq data into scRNA-seq data. The pre-defined gene activity matrix is often of low quality and does not reflect the dataset-specific relationship between the two data modalities. We propose scDART, a deep learning framework that integrates scRNA-seq and scATAC-seq data and learns cross-modalities relationships simultaneously. Specifically, the design of scDART allows it to preserve cell trajectories in continuous cell populations and can be applied to trajectory inference on integrated data.

Keywords: scATAC-seq, scRNA-seq, Trajectory inference, Integrative analysis, Single-cell multiomics

Background

The availability of single-cell multi-modality data provides a comprehensive view of every single cell. Single-cell RNA-Sequencing (scRNA-seq) and single-cell ATAC-Sequencing (scATAC-seq) respectively measure the gene expression and chromatin accessibility profiles of cells, each being considered as an important aspect of a cell. Recently, techniques which can measure both gene expression and chromatin accessibility in the same cells have been proposed [1–3], but these technologies are still not widely used, and they can suffer from low sensitivity of one of the data modalities. To make use of the enormous amount of existing data, computational methods have been proposed to integrate scRNA-seq and scATAC-seq data obtained separately for the same cell types in different batches [4–7], with the aim of building larger datasets and potentially learning the relationship between chromatin region and genes. Following a recent review paper on single cell data integration methods [8], we term scRNA-seq and scATAC-seq data that are not jointly profiled in the same cells as *unmatched data*, and integrating such datasets as the *diagonal integration task*. A diagonal integration method is expected to learn an integrated dataset in the form of either low-dimensional latent embedding or a high-dimensional integrated count matrix, where batch effects are removed and cell identity (e.g., the cluster membership) is preserved from the single-modality dataset to the integrated dataset.

A growing number of computational tools have been proposed for diagonal integration. Some methods aim to learn cell latent embedding such that the latent embedding can be used to reconstruct the original dataset [5, 6, 9]. Some use manifold alignment [10] and aim to learn cell latent embedding by enforcing the latent embedding to preserve the pairwise distances of cells in the original feature space (e.g., gene expression space, chromatin accessibility space) [11, 12]. `Seurat` (v3) [4] maps a query dataset to a reference dataset using canonical correlation analysis and obtains a new data matrix for the query dataset based on the reference dataset. Most of these methods integrate unmatched scRNA-seq and scATAC-seq datasets into the latent space that preserves the cluster structure in the original datasets, but they do not specifically accommodate the case where the cells form continuous trajectories instead of discrete clusters. When the cells form continuous trajectories instead of discrete clusters, the identity of a cell is the location of the cell along the trajectory. For example, if the trajectory has a structure of a rooted tree, the identity of a cell is reflected by both its branch membership and its pseudotime. Since each cell has a unique branch membership and pseudotime, preserving the cell's identity in a continuous population can be a more challenging task compared to that in discrete populations with clusters, where multiple cells share the same cluster identity.

On the other hand, a majority of the existing diagonal integration methods [4–6] require a pre-defined *gene activity matrix* (GAM, also called a region-gene association matrix), representing which genomic regions regulate the expression of which genes, to transform the scATAC-seq data into scRNA-seq data by multiplying the GAM to the scATAC-seq data matrix. The limitations of such practice are as follows: (1) A common way to obtain the GAM is to consider the relative locations between the regions and the gene bodies on the genome and assume that the regulating relationship exists only between regions and genes that are closely located [4, 5, 13]. However, such GAMs can be highly inaccurate as closely located regions and genes do not necessarily have regulatory relationships. (2) Simply multiplying the GAM to the scATAC-seq data to obtain scRNA-seq data makes an assumption of linear relationships between regions and genes in cells, which is often not true in biological systems.

Hereby we propose `scDART` (single cell Deep learning model for ATAC-Seq and RNA-Seq Trajectory integration), a scalable deep learning framework that embeds data modalities into a shared low-dimensional latent space that preserves cell trajectory structures in the original datasets. `scDART` is a diagonal integration method for unmatched scRNA-seq and scATAC-seq data, which is considered a more challenging task than other integration tasks [8]. It incorporates a neural network which encodes the nonlinear gene activity function that represents the relationships between chromatin regions and genes, named the *gene activity module*. `scDART` allows one to learn the latent space representations of the integrated data and the gene activity module at the same time. It can also take advantage of partial cell matching information as prior: that is, if we know certain cells in the scRNA-seq data should be matched with certain cells in the scATAC-seq data, `scDART` uses those cells as anchor cells that help obtain an improved integration, and we name this version of scDART as scDART-anchor. scDART can also be adapted to any two data modalities which have cross-modality interactions. Even though scDART and scDART-anchor were designed for cells that form continuous trajectories, they can also work for cells that form discrete clusters (with anchor information when necessary).

We have tested scDART on three real datasets: two unmatched datasets where the scRNA-seq and scATAC-seq data are not jointly profiled, and a matched dataset where both chromatin accessibility and gene-expression are measured simultaneously in the same cells. We also proposed a simulation procedure to simulate scRNA-seq and scATAC-seq data in the same cells and tested scDART on simulated datasets. The simulated datasets allow us to quantitatively evaluate the quality of integration and the learned gene activity function. We have compared scDART with existing diagonal integration methods including Liger [5], Seurat v3 [4], UnionCom [12], MMD-MA [11], and scJoint [14] on both real and simulated datasets. The results show that scDART learns a joint latent space for both data modalities that well preserve the cell developmental trajectories, and gene activity function encoding the relationship between chromatin regions and genes that is more accurate than current standard practice.

Results

Overview of scDART

The schematics of scDART are shown in Fig. 1a. The input of scDART is a scRNA-seq count matrix, a scATAC-seq data matrix, and a pre-defined GAM. The pre-defined GAM is obtained with a commonly used procedure based on genomic locations (see the "Methods" section) and serves as prior information for scDART to learn the gene activity function that more accurately represents the relationship between scATAC-seq

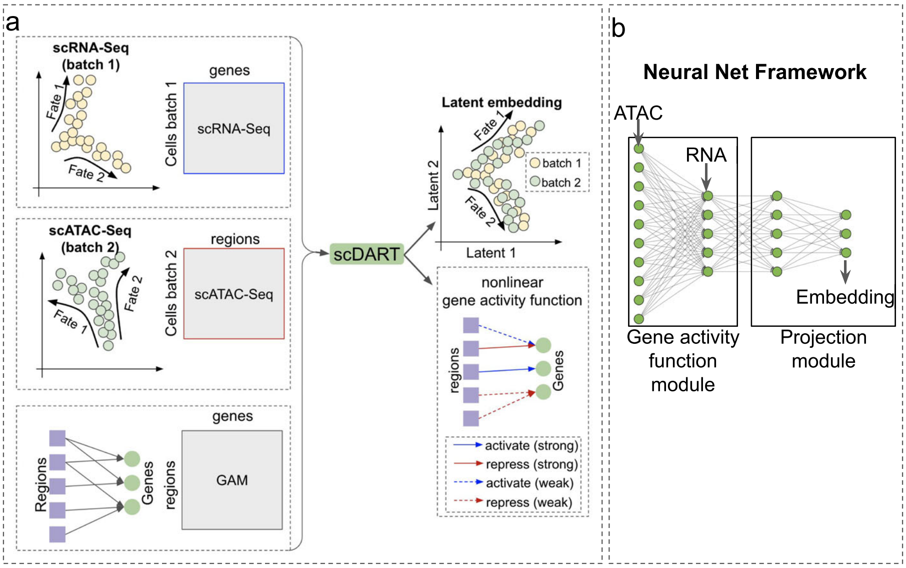

Fig. 1 Overview of scDART. **a** scDART takes as input a scRNA-seq data batch, a scATAC-seq data batch, and a pre-defined GAM. It learns the latent embedding of integrated data from the two data batches and a more accurate gene activity function between regions and genes. This gene activity function can be used to predict scRNA-seq data from scATAC-seq data (the predicted scRNA-seq data is also called pseudo-scRNA-seq data). **b** The neural network structure of scDART. scDART includes two modules: (1) the gene activity function module is a fully-connected neural network. This module encodes the nonlinear regulatory relationship between regions and genes, and generate the pseudo-scRNA-seq data from scATAC-seq data. (2) The projection module takes in the scRNA-seq data and the pseudo-scRNA-seq data and generates the latent embedding of both modalities

and scRNA-seq data. The pre-defined GAM is a binary matrix with rows corresponding to regions and columns corresponding to genes.

The neural network structure of scDART is shown in Fig. 1b. scDART consists of two modules: gene activity function module and projection module. The gene activity function module is a neural network that encodes the nonlinear gene activity function. It takes in the scATAC-seq matrix, transforms the chromatin accessibility of cells into gene expression, and generates a "pseudo-scRNA-seq" count matrix. The projection module takes in both scRNA-seq count matrix and the pseudo-scRNA-seq count matrix and projects them into a shared latent space.

The objective of scDART is designed considering three constraints. (1) To preserve cell identity and the trajectory structure in the latent space, scDART forces the pairwise Euclidean distances between cells in the latent space to approximate their relative distance along the trajectory manifold in the original feature space (gene expression and chromatin accessibility space). scDART uses *diffusion distance* to calculate cell relative distance on the trajectory manifold. *Diffusion distance* has been successfully used by trajectory inference methods [15, 16] to measure the differences between cells along the trajectory. It is advantageous in preserving the trajectory structure in the latent space compared to other distance metrics such as Euclidean or shortest-path distance [17]. In addition, *diffusion distance* can also be directly translated into the pseudotime of cells [15], which facilitates downstream analysis using integrated datasets such as trajectory inference and differential expression (DE) analysis. (2) We consider the scenario where cells in the two batches are sequenced from the same cell types; thus, they should have the same trajectory structure. scDART uses Maximum Mean Discrepancy (MMD) [18] to measure the similarity of the trajectory structures between the latent embedding of the cell batches, and minimizes the MMD loss such that the cells in different batches "merge" into the same trajectory. (3) scDART considers the pre-defined GAM as prior information to assist it to learn the gene activity function module which encodes a more accurate cross-modality relationship than the pre-defined GAM. A novel loss function is designed for scDART to incorporate this information.

We design the overall loss function considering all three constraints above. We denote the data matrix of scRNA-seq and scATAC-seq batches respectively as $\mathbf{X}_{\text{RNA}}$ and $\mathbf{X}_{\text{ATAC}}$, the latent embedding of scRNA-seq and scATAC-seq batches as $\mathbf{Z}_{\text{RNA}}$ and $\mathbf{Z}_{\text{ATAC}}$, and the parameters in the gene activity function module and projection module as $\mathbf{\Theta}_{\text{gact}}$ and $\mathbf{\Theta}_{\text{proj}}$. Then the overall loss function can be written as Eq. 1.

$$\begin{aligned} L = & L_{\text{dist}}\left(\mathbf{Z}_{\text{RNA}}, \mathbf{X}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}\right) + L_{\text{dist}}\left(\mathbf{Z}_{\text{ATAC}}, \mathbf{X}_{\text{ATAC}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}}\right) \\ & + \lambda_{\text{mmd}} \cdot L_{\text{mmd}}\left(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}}\right) + \lambda_g \cdot L_{\text{GAM}}\left(\mathbf{A}; \mathbf{\Theta}_{\text{gact}}\right) \end{aligned} \quad (1)$$

The first two loss terms, $L_{\text{dist}}(\mathbf{Z}_{\text{RNA}}, \mathbf{X}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}})$ and $L_{\text{dist}}(\mathbf{Z}_{\text{ATAC}}, \mathbf{X}_{\text{ATAC}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}})$, measure how well the pairwise Euclidean distances between cells in the latent embedding approximate the diffusion distances between cells, respectively in the scRNA-seq and the scATAC-seq batches. The third term, $L_{\text{mmd}}(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{RNA}})$, measures the difference between the distributions of cells in different batches in the latent space. Minimizing this term forces the two batches to "merge" into the same trajectory structure; thus, the batch effect which cause data matrices from different batches to deviate from each other is removed. The last term allows the gene activity module to incorporate the useful information in the pre-defined GAM. λ_{mmd} and λ_g are hyperparameters that control the strengths of the third and forth loss terms. A detailed explanation of each loss term is included in "Methods" section.

In certain cases, we have prior information on which cells from the two batches should have the same identity and should be aligned together. For example, the root cells of the trajectory in each batch of data are sometimes known in advance. scDART is able to use this information to obtain a better integration. When merging the two cell batches in the latent space, scDART takes in the root cells as the anchor cells and forces the anchor cells in two data batches to merge. The anchor-merge is achieved by adding an anchor loss term $L_{\text{anchor}}(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}})$ into Eq. 1. We term the version of scDART that uses the anchor cells as scDART-anchor. The detailed formulation of $L_{\text{anchor}}(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}})$ is included in "Methods" section.

The training process of scDART is in the "Methods" section. After having trained the model and obtained the latent embedding $\mathbf{Z}_{\text{ATAC}}$ and $\mathbf{Z}_{\text{RNA}}$, we apply a post-processing step to further refine the latent embedding to form a cleaner trajectory structure (see the "Methods" section for more details). The learned cell embedding can be used for trajectory inference and other downstream analyses. In this manuscript, we use Leiden clustering and minimum spanning tree (MST) to detect the trajectory backbones and DPT [15] to infer cell pseudotime (Methods).

scDART reconstructs cell trajectory and cell matching information in the mouse neonatal brain cortex dataset

To evaluate the performance of scDART, we tested it on a mouse neonatal brain cortex dataset obtained by SNARE-Seq [1], where the chromatin accessibility and gene expression profiles were jointly measured for every single cell. The dataset measured 1469 cells on the differentiation trajectory from intermediate progenitors to upper-layer excitatory neurons. We ran scDART assuming that chromatin accessibility and gene expression profiles are measured separately from two different cell batches, and evaluated how well scDART reconstructs the matching information between cells from the two batches.

We first visualized the latent space embedding of the integrated data learned with scDART and baseline methods (Fig. 2a, b for scDART, Seurat, Liger, and UnionCom results; Additional file 1: Fig. S1a for MMD-MA results and Additional file 1: Fig. S1b for scJoint results). Figure 2a shows the learned latent embedding of different methods, colored by cell types as annotated in the original paper [1]. In these plots, the expected cell trajectory which is a linear trajectory going through "IP-Hmg2"→"IP-Gadd45g"→"IP-Eomes"→"Ex23-Cntn2"→"Ex23-Cux1" should be preserved. To test this, we took the centroid of cells in each cell type (large red dots in the plots) and applied MST on these points using Euclidean distance between the centroids to obtain the trajectory backbone (red lines in plots). In Fig. 2a, only scDART clearly shows the expected trajectory. Figure 2b visualizes the integrated latent embedding colored with batch (or modality). In these plots, one expects to see that the two batches are merged and mixed. All methods merge the two batches except for Seurat. In Additional file 1: Fig. S1b, scJoint shows the linear trajectory but the two batches are not well mixed.

Based on the predicted trajectory and cell pseudotime from scDART, we analyzed the differentially expressed (DE) genes and motifs along the trajectory. Using the latent embedding of scDART, we inferred cell pseudotime on each modality separately and found genes that are DE with pseudotime using likelihood ratio test (see "Methods" for

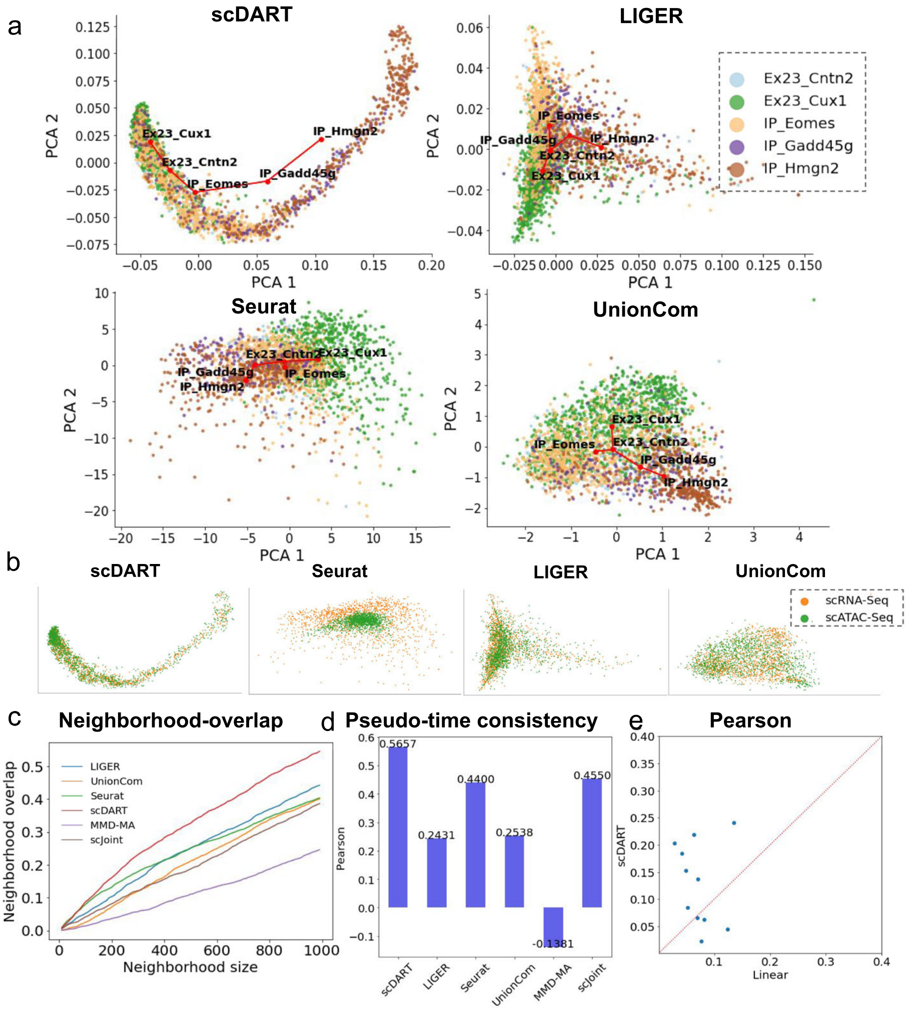

Fig. 2 The results of `scDART` and baseline methods on the SNARE-seq mouse neonatal brain cortex dataset. **a** Latent embedding of `scDART`, `Liger`, `Seurat`, and `UnionCom`, visualized using PCA. Cells are colored with cell type annotation in the original paper. Red lines show the inferred trajectory backbone. All plots share the same legend as in the `Liger` plot. **b** Latent embedding of `scDART`, `Seurat`, `Liger`, and `UnionCom`, where cells are colored with data batches. All plots share the same legend as in the `UnionCom` plot. **c** Neighborhood overlap score. **d** Pseudotime consistency score, where Pearson correlation is used. **e** Pearson correlation between real-scRNA-seq and (*y*-axis) pseudo-scRNA-seq (predicted scRNA-seq from scATAC-seq data by `scDART`), (*x*-axis) linear transformation. 11 key DE genes are shown

more details). Out of the DE genes we found, there are *Mki67* and *Fabp7* that are expected to be highly expressed in the initial stage of the trajectory [1], *Eomes* and *Unc5d* that are abundant in the neuroblast stages (their expression first increases then decreases), and *Cux1* and *Foxp1* that mark the upper-layer neurons (Additional file 1: Fig. S1c). These findings are consistent with the original paper [1].

We then transformed the chromatin accessibility into motif deviation using `ChromVAR` and detected the differentially accessible motifs along the trajectory (see the "Methods" section for more details). We found motif of transcription factor *Neurog1* ("MA0623.1_Neurog1"), a common regulator that involves in the initialization of neuron differentiation. We also found motif of transcription factors of SOX family such as *SOX6* ("MA0515.1_Sox6"), *SOX10* ("MA0442.1_SOX10"), and *SOX2* ("MA0143.3_Sox2"), which regulate the nervous system development (information from GeneCards [19]). The full lists of differentially expressed genes and accessible motifs are available in Additional file 2: Table S1.

Since this dataset has ground-truth cell matching information, we can quantify how well scDART integrates the two modalities in addition to the visualization in Fig. 2a, b, and compare it with baseline methods including Seurat, Liger, UnionCom, scJoint, and MMD-MA. We first calculate the *neighborhood overlap score*, which measures how many matched cells are located in the close neighborhood of each other in the latent space (see the "Methods" section for more details). The matched cells should be embedded into the exact same location as they have the same original cell identity. We calculate the neighborhood overlap score for different neighborhood sizes and plot the curve in Fig. 2c. Since the neighborhood overlap score has its limitations (that is, it considers only cell pairs in the same neighborhood, but if two matching cells are not in the same neighborhood, the distance between them is not considered), we used an additional measure, the cosine similarity score between every matching cell pair, which considers the similarity in the learned embedding of every matched cell pair across the two modalities (Methods). In Additional file 1: Fig. S2a, we observe that Liger and scDART have the highest cosine similarity score, whereas Seurat has a relatively low score. The results on the neighborhood overlap score and cosine similarity together show that scDART recovers cell matching better compared to other baseline methods.

In addition, we also quantify the consistency of pseudotime inferred from cells in each modality, since matched cells should ideally have the same pseudotime along the trajectory. We first infer the pseudotime of each data modality on the latent space separately, then calculate the correlation of pseudotime between matched cells using Pearson and Spearman correlation. The result shows that scDART has the best consistency and Seurat also achieves good performance close to scDART (Fig. 2d, Additional file 1: Fig. S2b). We further quantify the matching of latent embedding from different data modalities by running k-means clustering on both latent embedding and measuring the consistency of assigned clusters using Adjusted Rand Index (Methods). The result (Additional file 1: Fig. S2c) shows that scDART has the highest ARI. The comparisons between scDART and baseline methods through various metrics together show the superior performance of scDART on the dataset.

Finally, we can also use this jointly profiled dataset to test the non-linear gene activity function learned by scDART. Since for each cell in the dataset that is measured with scATAC-seq, we also know its corresponding scRNA-seq profile. We then compare the real scRNA-seq profile and the "pseudo-scRNA-seq" predicted from the gene activity module using the scATAC-seq data as input. We specifically investigated the DE genes along the pseudotime inferred with the gene expression modality. First, we visualize the predicted "pseudo-scRNA-seq" data and the real scRNA-seq data on these genes with heatmap (Additional file 1: Fig. S2d) and observe similar changing patterns of genes between the two plots.

We then compare the gene activity function of scDART with a linear transformation on the chromatin accessibility data using the input GAM in terms of the ability of predicting gene expression data from chromatin accessibility data. We quantify this ability by calculating Spearman and Pearson correlation between predicted gene expression data and measured gene expression data. From Fig. 2e and Additional file 1: Fig. S2e, f, we can observe that for most of the genes the correlation scores obtained by scDART are higher. We further compared scDART with Signac [20] that is also able to predict gene expression from chromatin accessibility. We measure the inference accuracy using Pearson correlation score of the top 50 DE genes along the trajectory, and the resulting boxplot (Additional file 1: Fig. S1d) shows that scDART predicts more accurate gene expression value compared to baseline methods. Despite the better performance of scDART over other baseline methods in predicting gene expression data from chromatin accessibility data, the accuracy of all methods is relatively low. This can be due to that the accessibility of a gene's promoting regions is insufficient to predict the expression level of the gene. In reality, the expression level of a gene can be affected by multiple factors including the expression level of its regulators.

scDART integrates mouse endothelial cell development datasets

We tested scDART on the mouse endothelial cell development dataset from [21]. The authors conducted scATAC-seq and scRNA-seq separately on two batches of mouse endothelial cells, where scATAC-seq measured one batch of 1186 cells and scRNA-seq measured another batch of 1628 cells. The cells undergo a differentiation path from endothelial cells (Endo) to hematopoietic stem and progenitor cells (HSPCs) that accumulate in intra-arterial clusters (IAC). The overall cell trajectory in this dataset is a linear-like path, where endothelial cells differentiate into "Arterial Endo 1/2" and then become "Pre-BN1" cells. "Pre-BN1" cells then undergo a maturation path to "IAC" through "Pre-HE" and "HE" stages.

We obtained the latent embedding of the integrated dataset using scDART and visualized it in Fig. 3a–c. We then applied our trajectory inference procedure (see the "Methods" section) to the latent embedding to infer the trajectory backbone (Fig. 3a) and cell pseudotime (Fig. 3c). The results (Fig. 3a–c) show that scDART is able to integrate scATAC-seq and scRNA-seq batches into the same latent space while preserving the linear trajectory structure. The sudden drop of the density of cells on the trajectory, where the cells are distributed densely at the beginning and then become very sparse at the end of the path, corresponds to the developmental "bottleneck" between "Pre-HE" and "HE" discussed in [21]. It also shows that there may be a "speeding up" of cell differentiation towards the end of the trajectory. We also performed baseline methods including Liger, Seurat, UnionCom, MMD-MA, and scJoint on the datasets and visualize their latent embedding using PCA (Fig. 3d, e, Additional file 1: Fig. S3a, b). Liger is able to merge the two batches of cells (Fig. 3e, top plot) but does not preserve the linear trajectory (Fig. 3d, top plot). Seurat shows a linear trajectory (Fig. 3d, middle plot) but cells from the two batches are not well integrated with the cells from scATAC-seq being more concentrated than those from the scRNA-seq batch (Fig. 3e, middle plot). UnionCom, while preserves the linear trajectory, mis-integrates "HE," "Pre-BN1," and "IAC" cell types (Fig. 3d, e, bottom plots). MMD-MA also fails to merge the cells from different batches (Additional file 1: Fig. S3a). scJoint captures the overall trajectory structure, but fails to integrate sub-population such as "pre-HE" cell type (Additional file 1: Fig. S3b).

We then analyzed the DE genes and differentially accessible motifs along the trajectory (see the "Methods" section for more details). We further performed gene enrichment analysis using TopGO [22] on the DE genes and found multiple enrichment terms relevant to hematopoietic stem and progenitor cells generation and other embryonic maturation processes. We found "myeloid cell differentiation," "positive regulation of

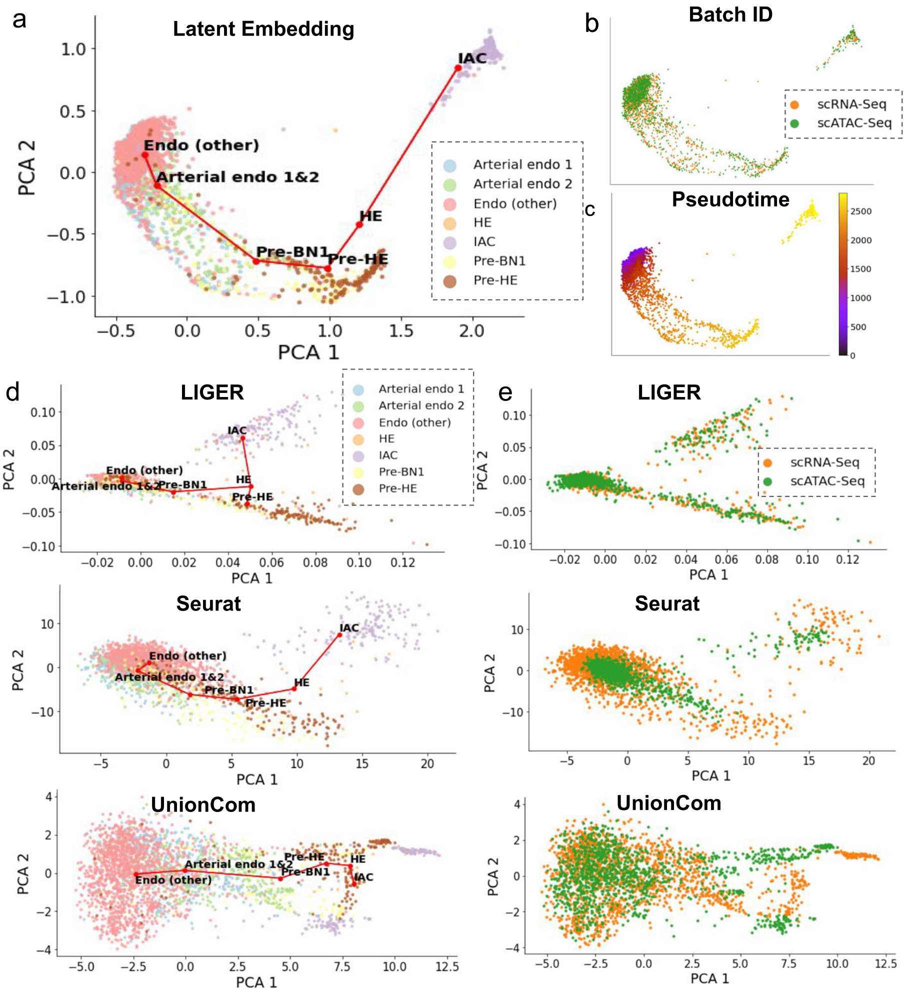

Fig. 3 The results of scDART and baseline methods on the mouse endothelial cell development dataset. **a–c** Latent embedding of scDART, visualized using PCA. Cells are colored with (**a**) cell type annotation from original data paper, (**b**) data batches, and (**c**) inferred pseudotime. Red lines in (**a**) show the inferred trajectory backbone. **d** The latent embedding of Seurat, Liger, and UnionCom where cells are colored with cell type annotation from original data paper. All plots share the same legend as in the Liger plot. **e** The latent embedding of Seurat, Liger, and UnionCom where cells are colored with data batches. All plots share the same legend as in the Liger plot

cell development," and "regulation of stem cell proliferation" in gene enrichment terms (Additional file 1: Fig. S3c, the full list is incorporated in Additional file 3: Table S2). Then we analyzed the motifs using ChromVAR and found enriched motifs using likelihood ratio test (see the "Methods" section for more details). Interestingly, we found multiple motifs of transcription factors including *RUNX1*, *GATA*, *SOX*, and *FOX*. Those TF motifs are also reported in the original paper to be closely related to hematopoietic stem and progenitor cells generation process (Additional file 1: Fig. S3d).

scDART integrates human hematopoiesis datasets and learns myeloid and lymphoid cell trajectories

We applied scDART to a human hematopoiesis dataset and analyzed the biological factors that drive the differentiation process from hematopoietic stem and progenitor cells (HSC) to myeloid and lymphoid cells. We collected scATAC-seq data from Buenrostro et al. [23] and scRNA-seq data from Pellin et al. [24]. 1367 cells from the scATAC-seq batch and

1666 cells from scRNA-seq batch were used. Myeloid and lymphoid cells are originated from HSC. HSC first develop into multipotent progenitors (MPP), then they undergo two potential differentiation branches until maturity: the Lymphoid-committed branch (CLP branch) and the Erythroid-committed branch (MEP branch). Cells in CLP branch first transit into Lymphoid multipotent progenitors (LMPP) and mature into common lymphoid progenitor cell (CLP), whereas cells in MEP branch undergo a differentiation path to megakaryocyte-erythroid progenitor (MEP) cells through common myeloid progenitor (CMP) cells. Note that in both papers presenting the original datasets [23, 24], the HSC and MPP cells were not separated and following these papers we use "HSC&MPP" to represent both cell types. Therefore, the expected trajectory on this dataset is a bifurcating trajectory with HSC&MPP as root cell type.

Figures 4a, b show the latent embedding of `scDART` and baseline methods. Cells from different data batches are well integrated by `scDART`. At the same time, the inferred trajectory on the latent embedding (backbone as the red lines in Fig. 4a, pseudotime in Additional file 1: Fig. S4a) also shows a clear bifurcating pattern following the differentiation path of HSC.

In terms of baseline methods, even though `Liger` and `UnionCom` can integrate cell batches, their latent embedding fail to show a correct trajectory that distinguishes myeloid and lymphoid cell branches (Fig. 4a, b). Both methods mistakenly assigned CMP as the branching point as the two branches were not sufficiently separated in their latent spaces. `Seurat`, `MMD-MA`, and `scJoint` have difficulty integrating cells from the two batches (Fig. 4a, b, Additional file 1: Fig. S4b, c), and `scJoint` does not detect the bifurcating structure.

We further analyzed the DE genes along both the MEP and CLP branches inferred from scRNA-seq data (see the "Methods" section for more details). We found DE genes such as *STMN1* and *GAPDH* in MEP branch, and *SATB1* and *RUNX2* in CLP branch. These genes was also shown to be the marker genes for the two branches in [24] (Fig. 4c). We conducted gene ontology enrichment analysis using `TopGO`. The top enriched terms (Additional file 1: Fig. S4d) include branch-specific terms such as myeloid leukocyte differentiation for MEP branch, and B cell receptor signaling pathway for CLP branch. We also find terms that are relevant to general cell differentiation process. We obtained the TF motifs deviation score from scATAC-seq data using `ChromVAR` and analyzed the differentially accessible motifs along both branches (see the "Methods" section for more details). We found motifs of GATA TF class ("MA0140.2_GATA1::TAL1," "MA0036.2_GATA2," "MA0037.2_GATA3," left column in Fig. 4d and top figure in Additional file 1: Fig. S4e) along MEP branch, which is the key regulator of MEP branch [23]. We also found motifs related to TFs such as EBP1 ("MA0154.3_EPB1"), TCF3 ("MA0522.2_TCF3"), TCF4 ("MAO830_TCF4") along the CLP branch (right column in Fig. 4d and bottom figure in Additional file 1: Fig. S4e). Those TFs were also reported in [23] as the key regulators of the CLP branch.

Testing scDART using simulated data

We proposed a simulation procedure which allows us to simulate scRNA-seq and scATAC-seq data in the same cells. Our simulation process considers that the chromatin accessibility data affects the probability that a gene's expression is switched *on* or *off*, and jointly simulates scRNA-seq and scATAC-seq data with this relationship (Additional

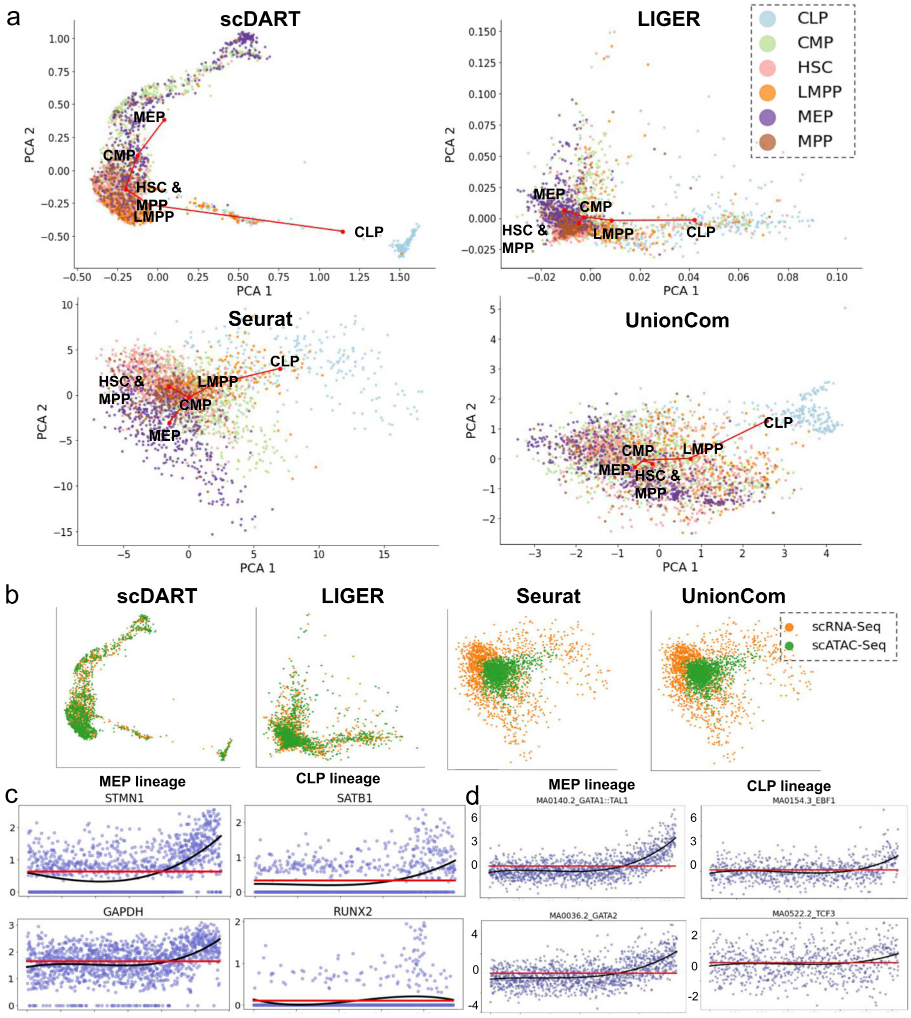

Fig. 4 The results of `scDART` and baseline methods on the human hematopoiesis dataset. **a** Latent embedding of `scDART`, `Liger`, `Seurat`, and `UnionCom`. Cells are colored with cell type annotations from the original paper. Red lines show the inferred trajectory backbone. All plots share the same legend as in the `Liger` plot. **b** Latent embedding of `scDART`, `Seurat`, `Liger`, and `UnionCom`. Cells colored with data batches. All plots share the same legend as in the `UnionCom` plot. **c** The expression level of *STMN1* and *GAPDH* along MEP lineage, and *SATB1* and *RUNX2* along CLP lineage. Cells are ordered on the x-axis according to the inferred pseudotime. **d** The deviation values (from ChromVAR) of differentially accessible motifs along MEP and CLP lineages. The black and red lines in (**c**) and (**d**) correspond to the fitted statistical models under alternative and null hypothesis, respectively, when conducting likelihood ratio test

file 1: Fig. S5a, Methods). We simulated multiple scRNA-seq and scATAC-seq datasets with different trajectory topology, as well as dataset with discrete clusters. Each dataset can have two or more batches, where both modalities are simulated for cells in every batch. From the simulated matched data, we can obtain unmatched data by keeping only one modality in one batch to test diagonal integration methods.

Quantifying latent embedding accuracy on simulated datasets

Using simulated data with continuous trajectories, we quantify the performance of `scDART` and `scDART-anchor`, along with baseline methods including `Liger`, `Seurat`, and `UnionCom`.

We first tested how well the cell latent embedding preserves the original trajectory structure. We simulated 6 datasets with different trajectory topologies: 3 bifurcating trajectory topology and 3 with trifurcating trajectory topology. The backbones of different trajectory structures used in simulation are shown in Fig. 5a. Using simulated data, we ran the diagonal integration methods and inferred trajectories on the latent embedding of each method (see the "Methods" section for details). Then, we measured the accuracy of cell branch assignment using F1 score [25]. We also measured how well the inferred pseudotime matches the true pseudotime along the trajectory using Kendall-τ score (also known as Kendall rank correlation). More details on F1 and Kendall-τ scores are in Methods. Since scDART and scDART-anchor include randomness when sampling the mini-batch in stochastic gradient descent, we ran them with 3 different random

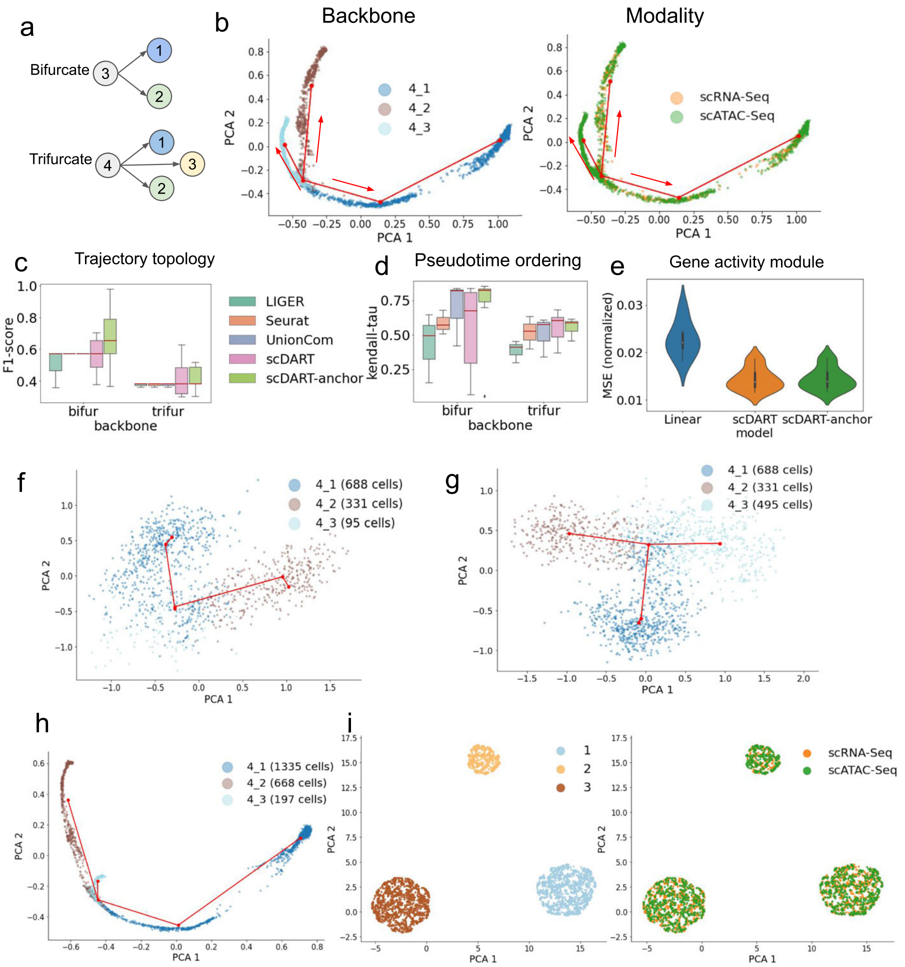

Fig. 5 Performance of scDART on simulated datasets. **a** Ground truth trajectory topology of simulated continuous datasets. **b** The latent embedding of scDART from one simulated data with trifurcating trajectory backbone. Red lines show the inferred backbone, and arrows show the trajectory direction. Left: cells colored by the ground truth trajectory branches they belong to; Right: cells colored by the data batches. **c, d** Boxplots of F1-score and Kendall-τ score calculated from different methods. 3 datasets were used for each type of trajectory, and for each dataset scDART and scDART-anchor were run 3 times. Seurat and UnionCom have the same F1 score on all 3 datasets with bifurcating trajectory. **e** Violin plots of normalized MSE between pseudo-scRNA-seq and ground truth scRNA-seq. **f** Trajectory backbone learned from scRNA-seq when branch 4_3 has only 95 cells. **g** Trajectory backbone learned from scRNA-seq when branch 4_3 has 495 cells. **h** Trajectory backbone learned from latent embedding integrated by scDART where when branch 4_3 has 197 cells (95 from scRNA-seq and 102 from scATAC-seq). **i** The latent embedding of scDART on one dataset with discrete clusters. Left: cells colored with cell type annotations; Right: cells colored with data batches

seeds for each dataset. When running scDART-anchor, we use 10 root cells with the smallest pseudotime from each dataset as the anchor.

One sample result of scDART is visualized in Fig. 5b. scDART is able to integrate the cells from two data batches into the latent space where the trifurcating trajectory pattern is preserved. The boxplots of scores are shown in Fig. 5c, d. Figure 5c shows that scDART learns latent embedding that is able to well preserve the trajectory topologies, especially when the root cell information is provided. Liger has the lowest F1 score mainly because the latent embedding of cells is not stretched apart enough along the cell pseudotime, which makes the trajectory inference algorithm detect the wrong branching structure. In Fig. 5d, scDART has some low Kendall-τ score for bifurcating topologies. The main reason is that the simulated data has almost the same cell density along the trajectory, which makes scDART mistakenly detect the wrong starting and ending point of those simple trajectory topologies. Such an equal-density scenario rarely happens in real dataset, as the cell does not always differentiate at the same speed, and it can be solved by using the anchor cell information. scDART-anchor stably achieves much higher F1 and Kendall-τ scores than other methods when using the root cell information.

Test gene activity module accuracy on simulated datasets

We use the gene activity module in scDART to encode the data-specific regulatory relationship between regions and genes, which meanwhile helps the model to learn a better cell embedding. We validate the capability of predicting scRNA-seq data from scATAC-seq data of this module on simulated data and compare the results with baseline procedure where the input GAM to scDART is used to for the prediction by linearly transforming the scATAC-seq data matrix into the scRNA-seq data matrix. Normalized mean square error between the predicted scRNA-seq data (also called pseudo-scRNA-seq data) and the ground truth scRNA-seq data was used as evaluation metric (see the "Methods" section for more details).

To prepare the data used for this test, we first generated N cells with jointly profiled scATAC-seq and scRNA-seq data using our simulation procedure, then divided these N cells into 2 batches with respectively N_1 and N_2 cells, and used the scRNA-seq data from batch 1 and the scATAC-seq data from batch 2 as the input for scDART. We take the output data from the gene activity module as the corresponding pseudo-scRNA-seq data for scATAC-seq data from batch 2 and compare it with the ground truth scRNA-seq data of batch 2.

The baseline method is to directly multiply GAM with the scATAC-seq data to obtain the pseudo-scRNA-seq data from batch 2, as is used in existing integration methods including Seurat and Liger. Note that with our simulated procedure we have true GAMs with binary values and we used the true GAMs for the baseline method. We termed the baseline method as *linear prediction*. We generated 6 simulated datasets for the test. The resulting violin plot (Fig. 5e) shows that scDART has a much lower error compared to *linear prediction* even when the true GAM was used in the *linear prediction*. scDART-anchor has a similar performance as scDART.

scDART detects branches with small number of cells

Performing integration of single cell data from more than one modality is expected to reveal knowledge that cannot be learned with single-modality data. On datasets where cells form discrete clusters, some methods were shown that they can detect rare cell types on the integrated dataset [4, 5, 26]. In the case of continuous trajectories, a branch in a trajectory may not be detected if the number of cells on that branch is very small in a dataset. Integrating this dataset with other datasets may help recover this branch. We show that scDART can detect such branch after integration.

Using our simulation procedure, we simulated one batch of scRNA-seq data and one batch of scATAC-seq data with a trifurcating ground truth trajectory where one branch is shorter than others and cells are also sparse on this branch (PCA visualization of scRNA-seq in Fig. 5f, and scATAC-seq in Additional file 1: Fig. S5b, where the small branch is "4_3"). Branch "4_3" cannot be detected when we apply the trajectory inference procedure (see the "Methods" section) on the scRNA-seq dataset after its dimension is reduced by PCA.(Fig. 5f). The branch cannot be detected with only scATAC-seq dataset either (Additional file 1: Fig. S5b, with the scATAC-seq data we used latent semantic indexing (LSI) for dimensionality reduction instead of PCA following [27, 28]). It is possible that applying a different dimensionality reduction method to the scRNA-seq data may allow branch "4_3" to be detected, but we show that the small number of cells on this branch is the main cause for the branch being undetected, by the results in Fig. 5g: the difference between the scRNA-seq datasets used in Fig. 5f and g lies in the number of cells on branch "4_3." For the former, there are 95 cells, and for the latter, there are 495 cells. Using the same trajectory inference procedure, branch "4_3" can be detected in Fig. 5g, and same for the case of scATAC-seq data (Additional file 1: Fig. S5b, c). After integrating the scRNA-seq and scATAC-seq data sets (scRNA-seq has 95 cells on the branch "4_3" and scATAC-seq has 107 cells on the branch "4_3") using scDART, branch "4_3" can be easily detected with the simple trajectory inference procedure (Fig. 5h and Additional file 1: Fig. S5d), which shows the capability of scDART in detecting small branches.

Test on imbalanced cell batches

In the tests above we used the simulated datasets where the number of cells is similar between batches, whereas in reality number of cells can vary significantly between batches. In order to test the performance of scDART on imbalanced cell batches, we generated a simulated dataset with trifurcating structure where there are 1486 cells in the scATAC-seq batch and only 757 cells in the scRNA-seq batch. We ran scDART on the simulated dataset. The result (Additional file 1: Fig. S6a, b) shows that scDART is able to integrate imbalanced cell batches.

Effects of hyper-parameters

The hyper-parameters in scDART (scDART-anchor) include the latent dimension d and the weights of loss terms in the loss function (Eq. 1), λ_g and λ_{mmd}. The latent dimension d is determined by the complexity of the trajectory structure in the data. The default value for d is 8. A larger d is needed for datasets with complex trajectory structures. λ_g controls how strong the prior gene activity matrix affects the training of the gene activity module. In all results we presented in this manuscript, we used the default value $\lambda_g = 1$. λ_g can be adjusted according to the quality of the input GAM — larger λ_g can be used if users have high confidence in the input GAM. λ_{mmd} controls how well the latent distributions of cells in different batches are "merged".

We comprehensively tested the robustness of scDART and scDART-anchor against different hyper-parameter settings using 4 simulated datasets selected from the datasets used in the previous tests. The datasets include both the bifurcating and trifurcating trajectories. For each dataset, we measured the F1 score and Kendall-τ score of our methods under different combinations of hyper-parameter values, where the values of each parameter are as follows: $\lambda_g = 0.1, 1, 10$, $\lambda_{mmd} = 1, 10$, and $d = 4, 8, 32$. For each dataset under each hyper-parameter setting, we run `scDART` and `scDART-anchor` with 3 different random seeds. The results are summarized as boxplots in Additional file 1: Fig. S6c, d. The results show that the F1 score lies within a stable range between 0.5 and 0.6, and the median Kendall-τ score also stays at around 0.6 under different hyper-parameter settings, which shows a robust performance of the model to hyper-parameters d, λ_{mmd} and λ_g. In addition, the comparison between the boxplots of `scDART` and `scDART-anchor` shows that higher robustness can be achieved when the anchor information is given.

scDART-anchor integrates multiple batches in each modality

`scDART` and `scDART-anchor` can be extended to the scenario where there are multiple batches in each modality through natural generalizations (Methods). We generated four batches of simulated data, consisting of 2 batches of scRNA-seq data and 2 batches of scATAC-seq data. The dataset has a bifurcating trajectory structure shown in Additional file 1: Fig. S7a. We use the root cells (10 cells at the beginning of the trajectory for each batch) as anchor cells. The learned latent embedding of four batches is shown in Additional file 1: Fig. S7a, b, where trajectories from all batches are well integrated.

Applying scDART and scDART-anchor to discrete populations

We also tested `scDART` and `scDART-anchor` on data from discrete clusters. We generated simulated datasets with discrete cluster structure (3 clusters, scRNA-seq batch has 1530 cells and scATAC-seq batch has 1470 cells) and used `scDART` to learn the latent embedding. The result (Fig. 5i) shows that `scDART` can also be applied to discrete clusters.

To apply our method to real data, we collected unmatched scRNA-seq and scATAC-seq data of mouse spleen dataset, where the scRNA-seq batch is from [29] and scATAC-Seq batch is from [30]. With this dataset, we apply `scDART-anchor`, by specifying one cluster from each batch which should match to each other. In practice, this prior information can be obtained by clustering and annotating the cell type of each cluster for each batch before integration. `scDART-anchor` only needs the information of one cluster in each batch which correspond to the same cell type. In Additional file 1: Fig. S8a, `scDART-anchor` used the cells from "T_CD4_naive" cluster as the anchor cells and is able to match all other cell types well. We also ran baseline methods on this dataset, including `Liger`, `Seurat`, and `UnionCom`. We visualized the cell embedding learned from different methods (Additional file 1: Fig. S8b-d) and quantified the performance using ARI score and graph connectivity score (see the "Methods" section). ARI score measures the separation of cells from different cell types in the integrated data. Graph connectivity score, on the other hand, measures how well cells from the same cell type are mixed across batches [31]. A good integration is expected to lead to high scores with both metrics. The barplots (Additional file 1: Fig. S8e, f) show that `scDART-anchor` has the highest graph connectivity score and the second highest ARI score. Seurat, even though has the highest ARI score, performs poorly in mixing the batches. We observe that `scDART-anchor` performs comparably to, if not better than, the baseline methods on this dataset.

Running time comparison

We further tested the scalability of scDART compared with the other baseline methods using simulated dataset. We generated one simulated dataset with 10,000 cells in total and tested the running time of the methods under different data sizes by sub-sampling the dataset into 500, 1000, 2000, 5000, and 10,000 cells. We run scDART, scJoint, and UnionCom on Nvidia A40 GPU and run Seurat and Liger on AMD 7452 CPU. The running time plot is shown in Additional file 1: Fig. S7c. The results show that scDART scales well with the increase of the data size, and can achieve comparable running time with Liger and scJoint when the number of cells is large.

Discussion

Through scDART, we have shown the effectiveness of learning the integrated data and cross-modality relationship simultaneously. Based on this idea, new methods can be developed to address some limitations in the current scDART model. First, like most existing methods, scDART is not designed for data batches where cells have different trajectories in different batches. Addressing disparities with continuous trajectories can be even more challenging than with discrete clusters, as disparities with continuous populations include various scenarios: additional or missing branches in the trajectories, different lengths of certain branches, different orders of cells on certain branches. We anticipate that the cross-modality relationship will play a more important role in methods integrating data with trajectory disparities.

Although the gene activity module learned in scDART can predict scRNA-seq data from scATAC-seq data more accurately compared to the conventional approach of linear transformation with GAM, the gene activity module is not perfect in performing this prediction task. In scDART, it has helped the integration task. To learn a highly accurate function that predicts scRNA-seq data from scATAC-seq data, matched (jointly-profiled) datasets can be leveraged for cell types where such data is available.

Currently, scDART works with two modalities, chromatin accessibility and gene expression. The framework can be generalized to work with three modalities, for example, chromatin accessibility, gene expression and protein abundance by adding another neural network representing the relationship between gene expression and protein abundance.

Conclusions

Although technologies which can jointly profile more than one modalities are available, a majority of the existing single-cell datasets are single-modality data and diagonal integration methods are needed to integrate different modalities from different batches. We proposed scDART, which is a diagonal integration method for scRNA-seq and scATAC-seq data, with the following advantages compared to existing methods: (1) existing methods use a pre-defined generic GAM to convert the scATAC-seq data into scRNA-seq data or map the manifold of the two data modalities without using the GAM. scDART learns the relationship between the scATAC-seq and the scRNA-seq data represented by a neural network, which is data-specific and can be nonlinear; (2) existing diagonal integration methods are heavily tested using datasets where cells form discrete clusters. scDART is particularly designed to preserve continuous trajectories in the integrated datasets and this strength of scDART has been shown by the comparison between scDART and existing methods using continuous populations of cells.

To our knowledge, scDART is the first method that performs the two important tasks, integrating two data modalities which are not jointly profiled and learning the cross-modality relationships, simultaneously in the case of scATAC-seq and scRNA-seq data. In the era of single cell multi-omics, the goal of data integration should be not only removing batch effects and compiling larger datasets, but also learning the relationship between different data modalities. We expect that more methods which can learn relationship across modalities will be developed in the future to take advantage of the multi-modal omics data.

Methods

The detailed loss terms in Eq. 1 are described in sections below. $L_{\text{dist}}(\cdot)$ is described in the section "Preserving trajectory structure in latent embedding"; $L_{\text{mmd}}(\cdot)$ is described in the section "Integrating modalities and batch removal"; $L_{\text{gact}}(\cdot)$ is described in the section "Using prior gene activity matrix." $L_{\text{anchor}}(\cdot)$ is described in the section "Using anchor information". In addition to the loss terms, the input GAM construction steps is described in the section "Constructing pre-defined gene activity matrix (GAM)". The model training procedure is described in the section "Training scDART". After training the model, the post-processing step is described in the section "Post-processing after training". The trajectory inference and differential expression that we used in the analysis above is described in the sections "Trajectory inference on the integrated latent space" and "Differential expression analysis". Simulated data generation procedure and the evaluation metric are described respectively in the sections "Data simulation" and "Evaluation metrics".

Constructing pre-defined gene activity matrix (GAM)

scDART constructs the pre-defined GAM as a binary matrix with rows corresponding to regions and columns corresponding to genes. scDART assumes the regions that lie within 2000 base-pairs upstream of the gene body on the genome to be the regulatory regions of that gene, and assign 1 to the corresponding elements in GAM, 0 to the remaining elements.

Preserving trajectory structure in latent embedding

The trajectory structure and the relative locations of cells on the trajectory can be represented by their pairwise diffusion distances calculated using their gene expression and chromatin accessibility features [15, 16]. In order to preserve trajectories in the latent space, we aim to minimize the difference between the pairwise distances in the latent space and in the original dimensional space. The pairwise distance between cells in the original space is calculated with diffusion distance and the pairwise distance in the latent space is calculated using Euclidean distance. We use L_{dist} to denote the difference between cells' Euclidean distance on the latent embedding and their diffusion distance and we would like to minimize L_{dist}. L_{dist} is calculated separately for scATAC-seq batch and scRNA-seq batch.

The calculation of the diffusion distance matrix is similar to [16]. Given a data matrix $\mathbf{X}$ (can be either the scRNA-seq or the scATAC-seq data matrix), we first reduce the feature dimension (using PCA for scRNA-seq and Latent Semantic Indexing for scATAC-seq). Then, we construct a pairwise similarity matrix $\mathbf{K}$ using their dimension-reduced representation. More specifically, the similarity between cells i and j (corresponding to the $(i,j)th$ element in $\mathbf{K}$) is calculated as:

$$\mathbf{K}(i,j) = \frac{1}{2}\exp\left(-\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma_i}\right)^{\alpha}\right) + \frac{1}{2}\exp\left(-\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma_j}\right)^{\alpha}\right) \tag{2}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the dimension-reduced representation of cell i and cell j. The bandwidth σ_i (σ_j) is set to be proportional to the distance between cell i (j) and its kth-nearest neighbor (k is set to 5). The decay parameter α is set to 40 (following default setting used in [16]). The cell transition matrix $\mathbf{P}$ is constructed by normalizing the similarity matrix $\mathbf{K}$ such that values in each row sum up to 1: $\mathbf{P} = \mathbf{K}/(\sum_j \mathbf{K}_{ij})$. Then, the diffusion process is performed by powering the transition matrix $\mathbf{P}$ to t times to obtain $\mathbf{P}_t = \mathbf{P}^t$. The diffusion step t is the key parameter in calculating the diffusion distance. Small t may not be enough to remove the noise in the dataset; large t, on the other hand, may remove noise and useful biological information at the same time. Inspired by [15] who summed up P_t of all t values from one to infinity to eliminate ts, we calculate the P_t with multiple t values ($t = 30, 50, 70$) selected from both small ts and large ts, and use the averaged $\bar{\mathbf{P}}_t$ following

$$\bar{\mathbf{P}}_t = \sum_{t=30,50,70} \frac{\mathbf{P}_t}{\|\mathbf{P}_t\|_2} \tag{3}$$

Finally, considering each row of $\mathbf{P}_t$ to be the feature vector of the corresponding cell, the diffusion distance matrix $\mathbf{D}_X$ can be calculated as the pairwise Euclidean distance between cells in this feature space. After calculating the diffusion distance matrix $\mathbf{D}_X$, we then calculate the Euclidean distance between every pair of cells using their latent embedding and obtain distance matrix $\mathbf{D}_Z$. L_{dist} then measures the difference between $\mathbf{D}_X$ and $\mathbf{D}_Z$ using KL-divergence:

$$L_{\text{dist}}(\mathbf{Z},\mathbf{X}) = KL(\mathbf{Q}_Z\|\mathbf{Q}_X) = \sum_{ij} \mathbf{Q}_Z(i,j)\log\frac{\mathbf{Q}_Z(i,j)}{\mathbf{Q}_X(i,j)} \tag{4}$$

where $\mathbf{Q}_X$ and $\mathbf{Q}_Z$ are normalized distribution matrix calculated from $\mathbf{D}_X$ and $\mathbf{D}_Z$:

$$\mathbf{Q}_X = \frac{\mathbf{D}_X}{\sum_{ij}\mathbf{D}_X(i,j)}; \quad \mathbf{Q}_Z = \frac{\mathbf{D}_Z}{\sum_{ij}\mathbf{D}_Z(i,j)} \tag{5}$$

Compared to other possible functions to measure the difference between $\mathbf{D}_X$ and $\mathbf{D}_Z$, for example, mean square error or inner product loss, the asymmetric formulation of KL-divergence loss (Eq. 4) has a larger penalty when $\mathbf{Q}_X(i,j)$ is small and $\mathbf{Q}_Z(i,j)$ is large, and this will force the latent embedding to better preserve the local manifold structure. This was also discussed in [32].

Integrating modalities and batch removal

Following existing work [12, 33], we assume the underlying trajectory structures of the input scRNA-seq and scATAC-seq datasets are similar as the cells follow the same biological process. In order to project scRNA-seq data and scATAC-seq data into the same latent space where the trajectory topology of both datasets merge, we incorporate the maximum mean discrepancy (MMD) loss (Eq. 6) [18]. MMD provides a statistical measure of the difference between the distributions of the latent embedding of scRNA-seq and scATAC-seq data. Denoting the latent embedding of scRNA-seq and scATAC-seq data respectively as $\mathbf{Z}_{\text{RNA}}$ and $\mathbf{Z}_{\text{ATAC}}$, the MMD loss function takes the following form:

$$L_{\text{mmd}}\left(\mathbf{Z}_{\text{RNA}}, \mathbf{Z}_{\text{ATAC}}, \gamma\right) = \mathbf{E}\left[K\left(\mathbf{Z}_{\text{RNA}}, \mathbf{Z}_{\text{RNA}}\right)\right] + \mathbf{E}\left[K\left(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{ATAC}}\right)\right] - 2\mathbf{E}\left[K\left(\mathbf{Z}_{\text{RNA}}, \mathbf{Z}_{\text{ATAC}}\right)\right] \quad (6)$$

where $\mathbf{E}$ means expectation; K is a Gaussian kernel function of the form:

$$K(\mathbf{Z}_1, \mathbf{Z}_2) = \exp\left(-\frac{\|\mathbf{Z}_1 - \mathbf{Z}_2\|_2^2}{2\gamma}\right) \quad (7)$$

γ is a key parameter of the Gaussian kernel function. Following [34], we sum up the MMD loss with different γ values to improve the robustness of the loss term:

$$L_{\text{mmd}}(\mathbf{Z}_{\text{RNA}}, \mathbf{Z}_{\text{ATAC}}) = \sum_{\gamma \in \Gamma} L_{\text{mmd}}(\mathbf{Z}_{\text{RNA}}, \mathbf{Z}_{\text{ATAC}}, \gamma) \quad (8)$$

where $\Gamma = \{10^u\}$ and u is an integer ranging from -6 to 6.

Using prior gene activity matrix

We denote the GAM that is obtained from the section *Constructing pre-defined gene activity matrix (GAM)* by $\mathbf{A}$. Some existing methods which integrate scRNA-seq and scATAC-seq data multiply $\mathbf{A}$ to the scATAC-seq data matrix and obtain another scRNA-seq data matrix [4–6], which is a linear transformation. However, such transformation is highly inaccurate. How the accessibility of a genomic region affects the expression level of a gene is a complex mechanism, which can be both nonlinear and cell-type specific.

We utilize a three-layer fully connected neural network, termed the “gene activity module,” to learn a gene activity function that can transform scATAC-seq data into scRNA-seq data. The gene activity module thus represents the data-specific relationship between scATAC-seq and scRNA-seq data, and it can encode nonlinearity in this relationship.

The network has an input dimension equal to the number of regions in scATAC-seq data, and an output dimension equal to the number of genes in scRNA-seq data. We use leaky rectified linear unit (ReLU_ℓ) [35] as the activation function between the layers and remove the bias term of each layer. Taking the region accessibility of a cell ($\mathbf{x}_{\text{ATAC}}$) as the input, the corresponding gene-expression data of that cell ($\mathbf{x}'_{\text{RNA}}$) can be predicted as

$$\mathbf{x}'_{\text{RNA}} = \mathbf{W}_3 \cdot (\text{Relu}_\ell(\mathbf{W}_2 \cdot (\text{Relu}_\ell(\mathbf{W}_1 \cdot \mathbf{x}_{\text{ATAC}})))) \quad (9)$$

where $\mathbf{W}_i$ represents the weights of the ith layer. When learning the weights, we use the coarse GAM $\mathbf{A}$ (which is a binary matrix) as prior information to constrain the training procedure. We assume that $\mathbf{A}$ includes all the potential regulations between regions and genes, that is, the 0s in $\mathbf{A}$ are correct information but 1s in $\mathbf{A}$ can be false positives. Given this assumption, we construct a regularization term L_{GAM} to penalize the non-zero regulation strength from a region to a gene in the learned gene activity module that should be zero according to $\mathbf{A}$:

$$L_{\text{GAM}} = \|(\prod_{i=1}^{\ell} \mathbf{W}_i) \odot \widehat{\mathbf{A}}\|_1 \quad (10)$$

$\odot$ denotes the element-wise multiplication between two matrices and $\widehat{\mathbf{A}}$ is the element-wise reversion of $\mathbf{A}$. We use ℓ_1 norm to enforce the sparsity of the learned regulation strength.

After finishing the training procedure and having learned $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{W}_3$, we then obtain a trained gene activity module which represents the complex nonlinear gene activity function between the scATAC-seq and the scRNA-seq data.

Using anchor information

Root cell information is often needed when performing trajectory inference algorithm. Such information can also be utilized in `scDART` as "anchor." That is, the root cells in the scRNA-seq dataset should be matched with the root cells in the scATAC-seq dataset. These cells are also called anchor cells. Other cells which are not root cells can also be anchor cells if we know their matching information across the two modalities. An additional loss term is added to the overall loss function when anchor cell information is used:

$$L_{\text{anchor}} = \|\bar{\mathbf{z}}_{\text{rna}}^{\text{anchor}} - \bar{\mathbf{z}}_{\text{atac}}^{\text{anchor}}\|_2^2$$

where $\bar{\mathbf{z}}_{\text{rna}}^{\text{anchor}}$ is the mean latent embedding of anchor cells within the scRNA-seq dataset, and $\bar{\mathbf{z}}_{\text{atac}}^{\text{anchor}}$ is the mean latent embedding of anchor cells within the scATAC-seq dataset. The loss make the anchor to match in the latent space by forcing the mean of the anchor cells' latent distribution to be closer. The weight for this loss term is 1.

Training scDART

When training `scDART`, the parameters in both the gene activity function module ($\mathbf{\Theta}_{\text{proj}}$) and the projection module ($\mathbf{\Theta}_{\text{gact}}$) are learned to minimize the overall loss function (Eq. 1). The training processes are different between scRNA-seq and scATAC-seq batches. When training with the scATAC-seq batch, we feed the data into the gene activity function module and take the transformed pseudo-scRNA-seq data into the projection module to obtain $\mathbf{Z}_{\text{ATAC}}$. We use stochastic gradient descent to update parameters in both modules and minimize Eq. 11 (part of Eq. 1 that is relevant to scATAC-seq batch).

$$\begin{aligned} L_{\text{ATAC}} = L_{\text{dist}}\left(\mathbf{Z}_{\text{ATAC}}, \mathbf{X}_{\text{ATAC}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}}\right) \\ + \lambda_{\text{mmd}} \cdot L_{\text{mmd}}\left(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}}\right) + \lambda_g \cdot L_{\text{GAM}}(\mathbf{A}, \mathbf{\Theta}_{\text{gact}}) \end{aligned} \tag{11}$$

When training with scRNA-seq batch, we directly feed the data into the projection module to obtain $\mathbf{Z}_{\text{RNA}}$. We use stochastic gradient descent to update parameters in only the projection module and minimize Eq. 12 (part of Eq. 1 that is relevant to scRNA-seq batch).

$$\begin{aligned} L_{\text{RNA}} = L_{\text{dist}}\left(\mathbf{Z}_{\text{RNA}}, \mathbf{X}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}\right) \\ + \lambda_{\text{mmd}} \cdot L_{\text{mmd}}\left(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}}\right) + \lambda_g \cdot L_{\text{GAM}}(\mathbf{A}, \mathbf{\Theta}_{\text{gact}}) \end{aligned} \tag{12}$$

The training of `scDART-anchor` follows the same procedure as `scDART`. The only difference is that `scDART-anchor` includes the anchor loss: $L_{\text{anchor}}(\mathbf{Z}_{\text{ATAC}}, \mathbf{Z}_{\text{RNA}}; \mathbf{\Theta}_{\text{proj}}, \mathbf{\Theta}_{\text{gact}})$ in both Eq. 11 and Eq. 12 when training on scATAC-seq and scRNA-seq batches.

Post-processing after training

After obtaining the latent embedding $\mathbf{Z}_{\text{ATAC}}$ and $\mathbf{Z}_{\text{RNA}}$, we apply a post-processing step to further refine the latent embedding to form a cleaner trajectory structure. We construct k ($k = 10$) mutual nearest neighbor graph [36] on the cells from $\mathbf{Z}_{\text{ATAC}}$ and $\mathbf{Z}_{\text{RNA}}$: for each cell in $\mathbf{Z}_{\text{RNA}}$, we find its k nearest cells in $\mathbf{Z}_{\text{ATAC}}$, and vice versa. After constructing the graph, we calculate weights on the graph. For cell i in $\mathbf{Z}_{\text{RNA}}$ and cell j in $\mathbf{Z}_{\text{ATAC}}$, the weight is:.

$$\mathbf{B}_{ij} = \exp\left(-\|\mathbf{z}_{\text{RNA}}(i) - \mathbf{z}_{\text{ATAC}}(j)\|_2^2\right) \tag{13}$$

then we update the latent embedding of each cell by the embedding of its neighbors. For example, for each cell i in the scRNA-seq data, the new latent embedding $\mathbf{z}'_{\text{RNA}}(i)$ is calculated as:

$$\mathbf{z}'_{\text{RNA}}(i) = \frac{\sum_{j\in\text{neigh}(i)} \mathbf{B}_{ij}\mathbf{z}_{\text{ATAC}}(j)}{\sum_{j\in\text{neigh}(i)} \mathbf{B}_{ij}} \tag{14}$$

The post-processing step makes different trajectory branches more distinguishable in the latent space, thus helping trajectory inference methods to detect more accurate trajectories in complex trajectory structures.

Extending scDART for multiple batches in each modality

`scDART` can be extended to integrate multiple data scRNA-seq batches and scATAC-Seq batches. During the training process, `scDART` takes all scATAC-Seq batches as input to the gene activity module and transforms them into batches of pseudo-scRNA-seq. Then `scDART` takes all batches of input scRNA-seq data and pseudo-scRNA-seq data into the projection module.

When calculating the loss function, a batch-specific L_{dist} loss is calculated for each batch of scRNA-seq data and each batch of scATAC-seq data, and the overall L_{dist} loss is the sum of all batch-specific L_{dist} loss terms. Before calculating each batch-specific L_{dist} loss term, the distance matrices of all batches are quantile normalized to reduce the differences in scale of the distance matrices between batches. The detailed procedure of quantile normalization is as follows: For two distance matrices where one is the reference matrix $\mathbf{D}_i$ and the other is target matrix $\mathbf{D}_j$, we first keep sampling values within $\mathbf{D}_i$ with replacement until the sampled values can fill up a matrix of the same size as $\mathbf{D}_j$, then we sort the sampled values and the values in $\mathbf{D}_j$. For each element in $\mathbf{D}_j$, we replace the value in it with the values in the sampled matrix of the same ranking after sorting.

The L_{mmd} loss is also generalized to merge all batches from all modalities together: we first select a reference batch (this can be any scRNA-seq batch, eg., we used the first scRNA-seq batch as reference batch) and calculate the MMD loss between each batch and the reference batch. Suppose there are in total b batches (including both scRNA-seq and scATAC-seq data), the extended MMD loss is:

$$L_{\text{mmd}}^{\text{multi}} = \sum_{i=\{1,\ldots,b\}, i\neq c} L_{\text{mmd}}(\mathbf{Z}_i, \mathbf{Z}_c)$$

where c is the index of the reference batch, and $\mathbf{Z}_i$ is the learned latent space representation of batch i.

When using scDART-anchor, the anchor loss is also calculated between the reference batch and each of other batches. The post-processing step of all batches is conducted in a pairwise manner. That is, the abovementioned post-processing procedure is repeated for each pair of batches.

Trajectory inference on the integrated latent space

scDART outputs latent space representations of the integrated data, and then any trajectory inference algorithm that takes the reduced dimensional space representation can be used to infer the cell trajectories, such as diffusion pseudotime (DPT) [15], Slingshot [37], and Monocle [38]. In our tests, we apply DPT [15] on the latent embedding to infer the pseudotime for cells from both modalities jointly. Our backbone inference procedure is similar to the procedure used in PAGA [39]. When inferring the trajectory backbone from the latent embedding, we first run Leiden clustering [40] on the latent embedding, then construct a fully connected graph on the cluster centroids with the pairwise Euclidean distance between cluster centroids as the weights of the edges between them, and run minimum spanning tree to infer the trajectory backbone on the cluster centroids.

Differential expression analysis

We find differentially expressed genes and accessible motifs along the trajectory by testing the significance of their changes depending on the pseudotime. We use likelihood ratio test as the significant test.

The alternative hypothesis assumes that the change of gene or motif depends on the pseudotime. We use a generalized additive model to fit their expression or accessibility values with pseudotime:

$$x \sim P(f(t)) \tag{15}$$

where $f(t)$ is build with degree-4 spline functions. For the link function $P(\cdot)$, we assume that the log-transformed gene expression and motif follow Gaussian distribution:

$$x_{\text{gene/motif}} \sim \text{Gaussian}(f(t)) \tag{16}$$

The null hypothesis assumes that:

$$x \sim P(c) \tag{17}$$

where c is a constant.

We then compare the two nested models using likelihood ratio test. We conduct the test for every gene and motif, and sort them separately according to their p-values. The significant genes and motifs are selected based on both their p-values and their relative ordering. We select the genes with p-values smaller than 0.05 and total number 100 cut-off and select the motifs with p-values smaller than 0.05 and total number 50 cut-off.

Data simulation

The simulated scRNA-seq and scATAC-seq data are generated with an extended version of SymSim [41] which simulates scRNA-seq data. In SymSim, a kinetic model is used to model the mRNA counts in cells, where a gene is considered to be either in an *on* state or in an *off* state [42]. When a gene is in the *on* state, its transcripts are synthesized with rate s, and synthesized mRNAs degrade with a rate d. A parameter k_{on} represents the rate at which a gene enters the *on* state, and k_{off} represents the rate of the gene entering the *off* state. To generate multiple discrete or continuous cell types, SymSim defines an "identity vector" for each cell, and the identity vectors can evolve along a user-provided tree which represents the trajectory backbone.

In this work, we extended SymSim so that it also generates scATAC-seq data. Additional file 1: Fig. S5a shows the process of generating N cells which have both scRNA-seq and scATAC-seq data. Denote the number of genes by G and the number of regions by R. A binary $R \times G$ GAM is provided to represent which regions affect which genes. As the scRNA-seq data depends on the scATAC-seq data, we first generate the scATAC-seq data. Similar to how SymSim generates scRNA-seq data along a continuous trajectory, we start with a "cell chromatin accessibility identity vector" of length v for the root cell and let it evolve along the given trajectory structure through a Brownian motion process to generate the "cell chromatin accessibility identity vectors" of cells along the tree. Each region has a "region identity vector" which is of the same length v. Multiplying the "cell chromatin accessibility identity matrix" and the "region identity vector matrix" we obtain an $N \times R$ matrix, where entries with larger values correspond to higher chromatin accessibility. We call this matrix "non-realistic scATAC-seq data" as its distribution is not the same as the distribution in real data. We then map the data in this matrix to a distribution obtained from a real scATAC-seq dataset [30] to get the realistic scATAC-seq data.

The scRNA-seq data is affected by both the input trajectory tree and the scATAC-seq data. We first generate the kinetic parameters for generating scRNA-seq data in the same way as in SymSim and obtain the "realistic kinetic parameter matrix" shown in Additional file 1: Fig. S5a. We now use the scATAC-seq data and the GAM to adjust k_{on}, as we consider that the accessibility of the associated regions of a gene affects the rate that the gene is switched on, which is what k_{on} corresponds to. Now among the three kinetic parameters of scRNA-seq data, k_{on} is affected by scATAC-seq data, and k_{off} and s are affected by the input trajectory; thus, we have combined both the effects of chromatin accessibility and cell differentiation process into the final scRNA-seq data. We then add technical noise to the scRNA-seq data and divide all cells into two batches while adding batch effects. To mimic the unmatched data, for one batch we keep only the scRNA-seq data and for the other batch we keep only the scATAC-seq data.

Evaluation metrics

When ground truth cell-cell correspondence information is available, we use the following metrics to evaluate the latent embedding learned by `scDART` and the trajectories inferred based on it: neighborhood overlap score, cosine similarity score, F1 score [25], Kendall-τ score [43], and ARI (adjusted Rand Index) score. With simulated data, we also evaluate the gene activity module learned by `scDART`. Given scATAC-seq data of a cell, we use the gene activity module of `scDART` to generate its pseudo-scRNA-seq data, and measure the normalized mean square error (MSE) between pseudo-scRNA-seq data and the ground truth scRNA-seq data of the cell.

Neighborhood overlap score [7, 12] can be used to measure how well datasets are integrated when there exists cell-cell correspondence across data modalities. Given a neighborhood size k, it constructs a k-nearest neighbor graph on the latent embedding of cells from both scRNA-seq and scATAC-seq data, and calculates the proportion of cells that have their corresponding cells in the other modality included within its neighborhood.

We further measure the recovery of cell-cell correspondence using cosine similarity score. For each cell, we calculate the cosine similarity score using its latent embedding from different modalities (Eq. 18 for score of cell i). Then, we average the cosine similarity score over all cells within the dataset, which correspond to the final cosine similarity score. A higher cosine similarity score shows a better recovery of cell-cell correspondence.

$$cos(\mathbf{Z}_{\text{RNA}}(i), \mathbf{Z}_{\text{ATAC}}(i)) = \frac{\mathbf{Z}_{\text{RNA}}(i) \cdot \mathbf{Z}_{\text{ATAC}}(i)}{\|\mathbf{Z}_{\text{RNA}}(i)\| \|\mathbf{Z}_{\text{ATAC}}(i)\|} \tag{18}$$

The latent embedding of the integrated data is evaluated through both visualization and the quantitative accuracy of the inferred trajectories. The accuracy of trajectory is measured from two different aspects: the accuracy of cell branch assignment, and the accuracy of cell pseudotime assignment. We measure cell branch assignment using F1 score which was used in [25]. Here we briefly describe the calculation of F1 score. Given the ground truth and inferred cell branch assignment, we first calculate the Jaccard similarity between every pair of inferred and ground truth cell branches. For every two cell branches, the Jaccard similarity is calculated as the size of their intersection cell sets over the size of their union cell sets. For every branch in ground truth or inferred trajectory, we calculate its "maximum Jaccard similarity" as the maximum value out of its Jacaard similarities with all branches in the inferred/ground truth trajectory. Then, we can calculate the *recovery* as the average maximum Jaccard similarity for every branch in ground truth and the *relevance* as the average maximum Jaccard similarity for every branch in inferred branches. The F1 score is then calculated as

$$F1 = 2/\left(\frac{1}{\text{recovery}} + \frac{1}{\text{relevance}}\right) \tag{19}$$

F1 score lies within the range between 0 and 1. The higher the score is, the better cell branches are assigned. We measure cell pseudotime assignment using Kendall-τ score [43], which is a rank-based correlation measurement that is commonly used to measure pseudotime inference accuracy [37, 44]. Kendall-τ score lies within the range between − 1 and 1. A higher Kendall-τ score means a better pseudotime inference accuracy.

We use an additional metric, ARI score, to measure the matching of latent embedding from scRNA-seq and scATAC-seq data given the ground truth cell-cell correspondence. First we run k-means clustering algorithm on the latent embedding of scRNA-seq data and scATAC-seq data separately, which will generate two cluster identity labels for each cell (one from the clustering of scRNA-seq, the other from clustering of scATAC-Seq). We measure the consistency between the two clustering results using Adjusted Rand Index (ARI) [45]. The number of clusters in k-means algorithm is set to be the number of ground truth cell types in the dataset.

In simulated datasets, we can retrieve the ground truth gene expression data for cells in the scATAC-seq batch. Then, we can measure how well the pseudo-scRNA-seq data learned from the gene activity module matches the ground truth scRNA-seq data using normalized MSE. Denoting the pseudo-scRNA-seq data of cell i as $\hat{\mathbf{X}}_i$, and the ground truth scRNA-seq data as $\mathbf{X}_i$, the normalized MSE is calculated as

$$\text{MSE}_{\text{norm}} = \frac{1}{N}\sum_{i=1}^{N} \left\| \frac{\mathbf{X}_i}{\|\mathbf{X}_i\|_2} - \frac{\hat{\mathbf{X}}_i}{\|\hat{\mathbf{X}}_i\|_2} \right\|_2^2 \tag{20}$$

where N is the total number of cells.

When evaluating the latent embedding of datasets with discrete cell type clusters, we use ARI score and graph connectivity score [31]. When calculating the ARI score, we first run Leiden clustering algorithm on the latent embedding with different cluster resolutions (from 0.1 to 1 with stepsize 0.5) and calculate the ARI scores using the ground truth cell type labels and the cluster labels under different resolutions. For each method, we select the highest ARI score as the final ARI score among all ARI scores obtained under different cluster resolutions. The final score is between 0 and 1. A higher score corresponds to a result with better cell type separation. We calculate the graph connectivity score following the procedures in [31]: we first construct a k-nearest neighbor graph from all cells using their latent embedding; Then, we extract the sub-graph for the cells from each cell type; We calculated the size of the largest connected component for each sub-graph; We normalize the largest connected component size and average it over all sub-graphs. The final score (between 0 and 1) quantifies the mixing of cell batches – a higher score corresponds to a better mixing result.

Settings of scDART and baseline methods

Setting of scDART

Before running `scDART`, we first filter genes and regions in the data matrix. For real datasets, we select first 500 or 1000 highly variable genes using SCANPY [46] and select the regions in scATAC-seq datasets that lie within the gene body or 2000 base-pairs upstream of the selected genes on the genome. For simulated datasets, we do not conduct feature filtering step. We further conduct library size normalization and log-transform on scRNA-seq data and binarize scATAC-seq data before feeding the data into `scDART`.

The hyper-parameters in `scDART` (`scDART-anchor`) include the latent dimension d and the weights of loss terms λ_g and λ_{mmd}. We used $d = 4$ for all real datasets and $d = 8$ for simulated continuous datasets as simulated datasets include bifurcating and trifurcating structures. Regarding λ_g, we used the default value $\lambda_g = 1$ in all results we presented in this manuscript. Regarding λ_{mmd}, in most of our test results we used the default value $\lambda_{mmd} = 1$, and only on the mouse neonatal brain cortex dataset we set $\lambda_{mmd} = 10$ for a stronger merging effect. We train `scDART` using Adam optimizer and ran the algorithm for 500 epochs. The parameter of network architecture is shown in Table 1, where Leaky ReLU with negative slop 0.2 is used as the activation function, and batch normalization is also used between layers.

Settings of baseline methods

We run `Seurat` following the pipeline in the online tutorial (https://satijalab.org/seurat/archive/v3.0/atacseq_integration_vignette.html) for the PBMC dataset, and use the same parameter setting as the one used in the tutorial. We used the default value of the number of principle components in function `FindTransferAnchors()` which is 30.

The key parameter in `Liger` is the latent space dimension. We set the latent space dimension to be the same as the ground truth number of cell types in real datasets. For simulated datasets, we set the latent space dimension of `Liger` to be 8.

We run `UnionCom` using the default hyper-parameters of the model and set the number of epochs to be 10000. The number of latent dimensions in the default setting is 32.

Table 1 Number of neurons at each layer of `scDART`, where n_{regions}, n_{genes}, and n_{latent} refer to number of regions, genes, and latent dimensions respectively

	Input dimension	Layer 1	Layer 2	Output dimensions
Gene activity module	n_{regions}	1024	512	n_{genes}
Projection module	n_{genes}	512	128	n_{latent}

scJoint takes as input the scRNA-seq count matrix, gene activity score matrix transformed from the scATAC-seq count matrix, and the cell type label of scRNA-seq data. We generate gene activity score matrix using the function in Seurat and select only the overlapped genes between raw scRNA-seq and gene activity score matrix. Then, we use the cell type label generated in the original data paper as the input for scJoint. We ran scJoint using different parameter settings (center_weight={1, 20, 50, 100}, with_crossentropy={True, False}, and embedding_size={64, 32}) and chose the results which look best according to the visualizations. The results shown used embedding_size=64.

When running MMD-MA, we calculate the similarity matrix by first reducing the feature dimension of scRNA-seq and scATAC-seq to 100 using PCA, and then calculating the inner product between cells using the reduced feature dimensions. We further set the $\lambda_1 = 10^{-6}$ and $\lambda_2 = 10^{-2}$, the latent dimension to be 2 in MMD-MA for visualization.

We ran Signac on mouse neonatal brain cortex dataset following the same pipeline in its online tutorial (https://satijalab.org/signac/articles/mouse_brain_vignette.html). We use the count matrix generated by "GeneActivity()" function with parameter "extend.upstream" equal to 2000 and "extend.downstream" equal to 0.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13059-022-02706-x.

Additional file 1. Supplementary figures S1-S8.

Additional file 2. Supplementary table S1. Table of .xlsx format. Differentially expressed genes and accessible motifs in mouse neonatal brain cortex dataset.

Additional file 3. Supplementary table S2. Table of .xlsx format. Gene ontology terms in mouse endothelial dataset.

Additional file 4. Review history.

Acknowledgements

The authors would like to thank Dr. Xi Chen (from SUSTech) for helpful discussions.

Review history

The review history is available as Additional file 4.

Peer review information

Stephanie McClelland was the primary editor of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Authors' contributions

ZZ and XZ conceived the project. ZZ and CY implement the model. XZ supervised the research. ZZ and XZ contributed to the writing of the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported in part by the US National Science Foundation DBI-2019771 and National Institutes of Health grant R35GM143070.

Availability of data and materials

scDART has been implemented as a Python package, which is freely available under the GPL-3 license. Source code has been deposited at the GitHub repository (https://github.com/PeterZZQ/scDART) [47]. The source code is also available at Zenodo repository [48]. The source code of the simulation tool that is used in this study has been deposited at the GitHub repository (https://github.com/PeterZZQ/Symsim2) [49], and is available at Zenodo repository [50]. The testing script has been deposited at the GitHub repository (https://github.com/PeterZZQ/scDART_test) [51]. The datasets analyzed in this study are available from the Gene Expression Omnibus (GEO) and ArrayExpress repository under the following accession numbers: GSE126074, GSE137117, GSE96772, GSE117498, E-MTAB-9769, and E-MTAB-6714.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 4 January 2022 Accepted: 14 June 2022
Published online: 27 June 2022

References

1. Chen S, Lake BB, Zhang K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. Nat Biotechnol. 2019;37(12):1452–57.
2. Cao J, Cusanovich DA, Ramani V, Aghamirzaie D, Pliner HA, Hill AJ, Daza RM, McFaline-Figueroa JL, Packer JS, Christiansen L, Steemers FJ, Adey AC, Trapnell C, Shendure J. Joint profiling of chromatin accessibility and gene expression in thousands of single cells. Science. 2018;361(6409):1380–85.
3. Ma S, Zhang B, LaFave LM, Earl AS, Chiang Z, Hu Y, Ding J, Brack A, Kartha VK, Tay T, Law T, Lareau C, Hsu Y-C, Regev A, Buenrostro JD. Chromatin potential identified by shared single-cell profiling of RNA and chromatin. Cell. 2020;183(4):1103–111620.
4. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM3rd, Hao Y, Stoeckius M, Smibert P, Satija R. Comprehensive integration of Single-Cell data. Cell. 2019;177(7):1888–190221.
5. Welch JD, Kozareva V, Ferreira A, Vanderburg C, Martin C, Macosko EZ. Single-cell multi-omic integration compares and contrasts features of brain cell identity. Cell. 2019;177(7):1873–188717.
6. Duren Z, Chen X, Zamanighomi M, Zeng W, Satpathy AT, Chang HY, Wang Y, Wong WH. Integrative analysis of single-cell genomics data by coupled nonnegative matrix factorizations. Proc Natl Acad Sci U S A. 2018;115(30): 7723–28.
7. Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, Baglaenko Y, Brenner M, Loh P-R, Raychaudhuri S. Fast, sensitive and accurate integration of single-cell data with harmony. Nat Methods. 2019;16(12):1289–96.
8. Argelaguet R, Cuomo ASE, Stegle O, Marioni JC. Computational principles and challenges in single-cell data integration. Nat Biotechnol. 2021;39(10):1202–15.
9. Argelaguet R, Arnol D, Bredikhin D, Deloro Y, Velten B, Marioni JC, Stegle O. MOFA+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biol. 2020;21(1):111.
10. Cui Z, Chang H, Shan S, Chen X. Generalized unsupervised manifold alignment. Adv Neural Inf Process Syst. 2014;27:2429–37.
11. Singh R, Demetci P, Bonora G, Ramani V, Lee C, Fang H, Duan Z, Deng X, Shendure J, Disteche C, Noble WS. Unsupervised manifold alignment for single-cell multi-omics data. In: Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. New York: Association for Computing Machinery; 2020. p. 1–10.
12. Cao K, Bai X, Hong Y, Wan L. Unsupervised topological alignment for single-cell multi-omics integration. Bioinformatics. 2020;36(Supplement_1):48–56.
13. Wang C, Sun D, Huang X, Wan C, Li Z, Han Y, Qin Q, Fan J, Qiu X, Xie Y, et al. Integrative analyses of single-cell transcriptome and regulome using maestro. Genome Biol. 2020;21(1):1–28.
14. Lin Y, Wu T-Y, Wan S, Yang JYH, Wong WH, Wang YXR. scjoint integrates atlas-scale single-cell RNA-seq and ATAC-seq data with transfer learning. Nat Biotechnol. 2022;40(5):703–10.
15. Haghverdi L, Büttner M, Wolf FA, Buettner F, Theis FJ. Diffusion pseudotime robustly reconstructs lineage branching. Nat Methods. 2016;13(10):845–48.
16. Moon KR, van Dijk D, Wang Z, Gigante S, Burkhardt DB, Chen WS, Yim K, van den Elzen A, Hirn MJ, Coifman RR, Ivanova NB, Wolf G, Krishnaswamy S. Visualizing structure and transitions in high-dimensional biological data. Nat Biotechnol. 2019;37(12):1482–92.
17. Tenenbaum JB, De Silva V, Langford JC. A global geometric framework for nonlinear dimensionality reduction. Science. 2000;290(5500):2319–23.
18. Dziugaite GK, Roy DM, Ghahramani Z. Training generative neural networks via maximum mean discrepancy optimization. arXiv preprint arXiv:1505.03906. 2015.
19. Stelzer G, Rosen N, Plaschkes I, Zimmerman S, Twik M, Fishilevich S, Stein TI, Nudel R, Lieder I, Mazor Y, Kaplan S, Dahary D, Warshawsky D, Guan-Golan Y, Kohn A, Rappaport N, Safran M, Lancet D. The GeneCards suite: From gene data mining to disease genome sequence analyses. Curr Protoc Bioinformatics. 2016;54:1–30113033.
20. Stuart T, Srivastava A, Madad S, Lareau C, Satija R. Single-cell chromatin state analysis with signac. Nat Methods. 2021. https://doi.org/10.1038/s41592-021-01282-5.
21. Zhu Q, Gao P, Tober J, Bennett L, Chen C, Uzun Y, Li Y, Howell ED, Mumau M, Yu W, He B, Speck NA, Tan K. Developmental trajectory of prehematopoietic stem cell formation from endothelium. Blood. 2020;136(7):845–56.
22. Alexa A, Rahnenführer J. topgo: Enrichment analysis for gene ontology. R package version 2.44.0. 2021. https://doi.org/10.18129/B9.bioc.topGO.
23. Buenrostro JD, Corces MR, Lareau CA, Wu B, Schep AN, Aryee MJ, Majeti R, Chang HY, Greenleaf WJ. Integrated single-cell analysis maps the continuous regulatory landscape of human hematopoietic differentiation. Cell. 2018;173(6):1535–48.
24. Pellin D, Loperfido M, Baricordi C, Wolock SL, Montepeloso A, Weinberg OK, Biffi A, Klein AM, Biasco L. A comprehensive single cell transcriptional landscape of human hematopoietic progenitors. Nat Commun. 2019;10(1):1–15.
25. Saelens W, Cannoodt R, Todorov H, Saeys Y. A comparison of single-cell trajectory inference methods. Nat Biotechnol. 2019;37(5):547–54.
26. Johansen N, Quon G. scAlign: a tool for alignment, integration, and rare cell identification from scRNA-seq data. Genome Biol. 2019;20(1):1–21.

27. Cusanovich DA, Hill AJ, Aghamirzaie D, Daza RM, Pliner HA, Berletch JB, Filippova GN, Huang X, Christiansen L, DeWitt WS, et al. A single-cell atlas of in vivo mammalian chromatin accessibility. Cell. 2018;174(5):1309–24.
28. Satpathy AT, Granja JM, Yost KE, Qi Y, Meschi F, McDermott GP, Olsen BN, Mumbach MR, Pierce SE, Corces MR, et al. Massively parallel single-cell chromatin landscapes of human immune cell development and intratumoral t cell exhaustion. Nat Biotechnol. 2019;37(8):925–36.
29. Jain MS, Polanski K, Conde CD, Chen X, Park J, Mamanova L, Knights A, Botting RA, Stephenson E, Haniffa M, Lamacraft A, Efremova M, Teichmann SA. MultiMAP: dimensionality reduction and integration of multimodal data. Genome Biol. 2021;22(1):346.
30. Chen X, Miragaia RJ, Natarajan KN, Teichmann SA. A rapid and robust method for single cell chromatin accessibility profiling. Nat Commun. 2018;9(1):5345.
31. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Müller MF, Strobl DC, Zappia L, Dugas M, Colomé-Tatché M, et al. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods. 2022;19(1):41–50.
32. Van der Maaten L, Hinton G. Visualizing data using t-sne. J Mach Learn Res. 2008;9(86):2579–605. http://jmlr.org/papers/v9/vandermaaten08a.html.
33. Welch JD, Hartemink AJ, Prins JF. MATCHER: manifold alignment reveals correspondence between single cell transcriptome and epigenome dynamics. Genome Biol. 2017;18(1):138.
34. Amodio M, van Dijk D, Srinivasan K, Chen WS, Mohsen H, Moon KR, Campbell A, Zhao Y, Wang X, Venkataswamy M, Desai A, Ravi V, Kumar P, Montgomery R, Wolf G, Krishnaswamy S. Exploring single-cell data with deep multitasking neural networks. Nat Methods. 2019;16(11):1139–45.
35. Xu B, Wang N, Chen T, Li M. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853. 2015.
36. Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol. 2018;36(5):421–27.
37. Street K, Risso D, Fletcher RB, Das D, Ngai J, Yosef N, Purdom E, Dudoit S. Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genomics. 2018;19(1):477.
38. Qiu X, Mao Q, Tang Y, Wang L, Chawla R, Pliner HA, Trapnell C. Reversed graph embedding resolves complex single-cell trajectories. Nat Methods. 2017;14(10):979–82.
39. Wolf FA, Hamey FK, Plass M, Solana J, Dahlin JS, Göttgens B, Rajewsky N, Simon L, Theis FJ. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. Genome Biol. 2019;20(1):59.
40. Traag VA, Waltman L, van Eck NJ. From louvain to leiden: guaranteeing well-connected communities. Sci Rep. 2019;9(1):5233.
41. Zhang X, Xu C, Yosef N. Simulating multiple faceted variability in single cell RNA sequencing. Nat Commun. 2019;10(1):2611.
42. Munsky B, Neuert G, van Oudenaarden A. Using gene expression noise to understand gene regulation. Science. 2012;336(6078):183–87.
43. Kendall MG. A New Measure of Rank Correlation. Biometrika. 30(1/2):81. https://doi.org/10.2307/2332226, https://doi.org/10.2307%2F2332226.
44. Zhang Z, Zhang X. Inference of high-resolution trajectories in single-cell rna-seq data by using rna velocity. Cell Rep Methods. 2021;1(6):100095. https://doi.org/10.1016/j.crmeth.2021.100095.
45. Hubert L, Arabie P. Comparing partitions. J Classif. 1985;2(1):193–218.
46. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19(1):15.
47. Zhang Z, Yang C, Zhang X. Integrating unmatched scrna-seq and scatac-seq data and learning cross-modality relationship simultaneously. 2022. GitHub, https://github.com/PeterZZQ/scDART. Accessed 20 June 2022.
48. Zhang Z, Yang C, Zhang X. Integrating unmatched scrna-seq and scatac-seq data and learning cross-modality relationship simultaneously. 2022. GitHub, https://github.com/PeterZZQ/scDART_test. Accessed 20 June 2022.
49. Zhang Z, Yang C, Zhang X. Integrating unmatched scrna-seq and scatac-seq data and learning cross-modality relationship simultaneously. 2022. GitHub, https://github.com/PeterZZQ/Symsim2. Accessed 20 June 2022.
50. Zhang Z, Yang C, Zhang X. Integrating unmatched scrna-seq and scatac-seq data and learning cross-modality relationship simultaneously. 2022. Zenodo, https://doi.org/10.5281/zenodo.6600739.
51. Zhang Z, Yang C, Zhang X. Integrating unmatched scrna-seq and scatac-seq data and learning cross-modality relationship simultaneously. 2022. Zenodo, https://doi.org/10.5281/zenodo.6599946.

Publisher's Note