MIDAS: a deep generative model for mosaic integration and knowledge transfer of single-cell multimodal data

Zhen He[1,#] **Yaowen Chen**[1,#] **Shuofeng Hu**[1,#] **Sijing An**[1] **Junfeng Shi**[2] **Runyan Liu**[1] **Guohua Dong**[1] **Jinhui Shi**[1] **Jiaxin Zhao**[1] **Jing Wang**[1] **Yuan Zhu**[2] **Xiaochen Bo**[3,*] **Xiaomin Ying**[1,*]

[1]Center for Computational Biology, Beijing Institute of Basic Medical Sciences, Beijing, China
[2]School of Automation, China University of Geosciences, Wuhan, China
[3]Institute of Health Service and Transfusion Medicine, Beijing, China
[#]These authors contributed equally: Zhen He, Yaowen Chen, Shuofeng Hu

ABSTRACT

Rapidly developing single-cell multi-omics sequencing technologies generate increasingly large bodies of multimodal data. Integrating multimodal data from different sequencing technologies, *i.e.* mosaic data, permits larger-scale investigation with more modalities and can help to better reveal cellular heterogeneity. However, mosaic integration involves major challenges, particularly regarding modality alignment and batch effect removal. Here we present a deep probabilistic framework for the mosaic integration and knowledge transfer (MIDAS) of single-cell multimodal data. MIDAS simultaneously achieves dimensionality reduction, imputation, and batch correction of mosaic data by employing self-supervised modality alignment and information-theoretic latent disentanglement. We demonstrate its superiority to other methods and reliability by evaluating its performance in full trimodal integration and various mosaic tasks. We also constructed a single-cell trimodal atlas of human peripheral blood mononuclear cells (PBMCs), and tailored transfer learning and reciprocal reference mapping schemes to enable flexible and accurate knowledge transfer from the atlas to new data.

1 Introduction

Recently emerged single-cell multimodal omics (scMulti-omics) sequencing technologies enable the simultaneous detection of multiple modalities, such as RNA expression, protein abundance, and chromatin accessibility, in the same cell [1]. These technologies, including the trimodal DOGMA-seq [2] and TEA-seq [3] and bimodal CITE-seq [4] and ASAP-seq [2], among many others [5–10], reveal not only cellular heterogeneity at multiple molecular layers, enabling more refined identification of cell characteristics, but also connections across omes, providing a systematic view of ome interactions and regulation at single-cell resolution. The involvement of more measured modalities in analyses of biological samples increases the potential for breakthroughs in the understanding of mechanisms underlying numerous processes, including cell functioning, tissue development, and disease occurrence. The growing size of scMulti-omics datasets necessitates the development of new computational tools to integrate massive high-dimensional data generated from different sources, thereby facilitating more comprehensive and reliable downstream analysis for knowledge mining [1, 11]. Such "integrative analysis" also enables the construction of a large-scale single-cell multimodal atlas, which is urgently needed to make full use of publicly available single-cell multimodal data. Such an atlas can serve as an encyclopedia allowing researchers' transfer of knowledge to their new data and in-house studies [12–14].

Several methods for single-cell multimodal integration have been proposed recently. MOFA [15] and WNN [13] have been proposed for the integration of bimodal data with complete modalities, and totalVI [16], sciPENN [17], Cobolt [18], and MultiVI [19] are designed for bimodal integration with missing modalities. Fewer trimodal integration methods have been developed; MOFA+ [20] has been proposed for trimodal integration with complete modalities, and

*Correspondence to: Xiaomin Ying (email: `yingxmbio@foxmail.com`) and Xiaochen Bo (email: `boxiaoc@163.com`).

GLUE [21] and uniPort [22] were developed for the integration of unpaired trimodal data (*i.e.*, datasets involving single specific modalities).

All of these current integration methods are designed to handle specific and limited omics combinations. Due to the diversity of scMulti-omics technologies, datasets from different studies often include heterogeneous omics combinations with one or more missing modalities, resulting in a mosaic-like data. The mosaic-like data is increasing rapidly and is predictably prevalent. Thus, a flexible general integration method is urgently needed to markedly expand the scale and modalities of integration, breaking through the modality scalability and cost limitations of existing scMulti-omics sequencing technologies. However, multimodal mosaic integration is quite challenging [23, 24]. One major challenge is the reconciliation of modality heterogeneity and technical variation across batches. Another is the achievement of modality imputation and batch correction for downstream analysis.

To overcome these challenges, we developed a probabilistic framework, MIDAS, for the mosaic integration and knowledge transfer of single-cell multimodal data. By employing self-supervised learning [25] and information-theoretic approaches [26], MIDAS simultaneously achieves modality alignment, imputation, and batch correction for single-cell trimodal mosaic data. We further designed transfer learning and reciprocal reference mapping schemes tailored to MIDAS to enable knowledge transfer. Systematic benchmarks and case studies demonstrate that MIDAS can accurately and robustly integrate mosaic datasets. Through the atlas-level mosaic integration of trimodal human peripheral blood mononuclear cell (PBMC) data, MIDAS achieved flexible and accurate knowledge transfer for various types of unimodal and multimodal query datasets.

2 Results

2.1 MIDAS enables the mosaic integration and knowledge transfer of single-cell multimodal data

MIDAS is a deep generative model [27, 28] that represents the joint distribution of incomplete single-cell multimodal data with Assay for Transposase-Accessible Chromatin (ATAC), RNA, and Antibody-Derived Tags (ADT) measurements. MIDAS assumes that each cell's multimodal measurements are generated from two modality-agnostic and disentangled latent variables—the biological state (*i.e.*, cellular heterogeneity) and technical noise (*i.e.*, unwanted variation induced by single-cell experimentation)—through deep neural networks [29]. Its input consists of a mosaic feature-by-cell count matrix comprising different single-cell samples (batches), and a vector representing the cell batch IDs (Fig. 1a). The batches can derive from different experiments or be generated by the application of different sequencing techniques (*e.g.*, scRNA-seq [30], CITE-seq [4], ASAP-seq [2], and TEA-seq [3]), and thus can have different technical noise, modalities, and features. The MIDAS output comprises biological state and technical noise matrices, which are the two low-dimensional representations of different cells, and an imputed and batch-corrected count matrix in which modalities and features missing from the input data are interpolated and batch effects are removed. These outputs can be used for downstream analyses such as clustering, differential expression analyses, and cell typing [31].

MIDAS is based on a Variational Autoencoder (VAE) [32] architecture, with a modularized encoder network designed to handle the mosaic input data and infer the latent variables, and a decoder network that uses the latent variables to seed the generative process for the observed data (Fig. 1b). It uses self-supervised learning to align different modalities in latent space, improving cross-modal inference in downstream tasks such as imputation and translation (Fig. 1b). Information-theoretic approaches are applied to disentangle the biological state and technical noise, enabling further batch correction (Fig. 1b). Combining these elements into our optimization objective, scalable learning and inference of MIDAS are simultaneously achieved by the Stochastic Gradient Variational Bayes (SGVB) [33], which also enables large-scale mosaic integration and atlas construction of single-cell multimodal data. For the robust transfer of knowledge from the constructed atlas to query datasets with various modality combinations, transfer learning and reciprocal reference mapping schemes were developed for the transfer of model parameters and cell labels, respectively (Fig. 1c).

2.2 MIDAS shows superior performance in trimodal integration with complete modalities

To compare MIDAS with the state-of-the-art methods, we evaluated the performance of MIDAS in trimodal integration with complete modalities, a simplified form of mosaic integration, as few methods are designed specifically for trimodal mosaic integration. We named this task "rectangular integration". We used two (DOGMA-seq [2] and TEA-seq [3]) published single-cell trimodal human PBMC datasets with simultaneous RNA, ADT, and ATAC measurements for each cell to construct dogma-full and teadog-full datasets. Dogma-full took all four batches (LLL_Ctrl, LLL_Stim, DIG_Ctrl, and DIG_Stim) from the DOGMA-seq dataset, and teadog-full took two batches (W1 and W6) from the TEA-seq dataset and two batches (LLL_Ctrl and DIG_Stim) from the DOGMA-seq dataset (Supplementary Table 1).

The integration of each dataset requires the handling of batch effects and missing features and preservation of biological signals, which is challenging, especially for the teadog-full dataset, as the involvement of more datasets amplifies biological and technical variation.

Uniform manifold approximation and projection (UMAP) [34] visualization showed that the biological states of different batches were well aligned and that their grouping was consistent with the ground-truth cell types (Fig. 2a left), and that the technical noise was grouped by batch and exhibited little relevance to cell types (Fig. 2b). Thus, the two inferred latent variables were disentangled well and independently represented biological and technical variation.

Taking the inferred biological states as low-dimensional representations of the integrated data, we compared the performance of MIDAS with that of nine strategies derived from recently published methods (Methods) in the removal of batch effects and preservation of biological signals. UMAP visualization of the integration results showed that MIDAS ideally removed batch effects and meanwhile preserved cell type information on both dogma-full and teadog-full datasets, whereas the performance of other strategies was not satisfactory. For example, BBKNN+average, MOFA+, PCA+WNN, Scanorama-embed+WNN, and Scanorama-feat+WNN did not mix different batches well, and PCA+WNN and Scanorama-feat+WNN produced cell clusters largely inconsistent with cell types (Fig. 2a).

In a quantitative evaluation of the low-dimensional representations of different strategies performed with the widely used single-cell integration benchmarking (scIB) [35] tool, MIDAS had the highest batch correction, biological conservation, and overall scores for the dogma-full and teadog-full datasets (Fig. 2c and Supplementary Fig. 1). In addition, MIDAS preserved cell type-specific patterns in batch-corrected RNA, ADT, and ATAC data (Methods). For each cell type, fold changes in gene/protein abundance and chromatin accessibility in raw and batch-corrected data correlated strongly and positively (all $r > 0.8$; Fig. 2d).

Manual cell clustering and typing based on the integrated low-dimensional representations and batch-corrected data from MIDAS led to the identification of 13 PBMC types, including B cells, T cells, dendritic cells (DCs), natural killer (NK) cells, and monocytes (Fig. 2e). We got a distinct T cell cluster that highly expresses CD4 and CD8 simultaneously. We labeled this cluster as double positive (DP) $CD4^+$/$CD8^+$ T cells. This phenomenon was also reported in previous studies [36]. Another T cell cluster, containing mucosa-associated invariant T cells and gamma-delta T cells, was distinct from conventional T cells and was labeled as unconventional T cells [37].

As is known, multiple omes regulate biological functions synergistically [1]. MIDAS integrates RNA, ADT and ATAC single-cell data and hence facilitates to discover the intrinsic nature of cell activities in a more comprehensive manner. We found that all omics contributed greatly to the identification of cell types and functions.

Systematic screening for expression inconsistencies between proteins and their corresponding genes, expected to reflect ome irreplaceability, at the RNA and ADT levels demonstrated that several markers in each cell type were expressed strongly in one modality and weakly in the other (Fig. 2f, Supplementary Fig. 2). For instance, *MS4A1*, which encodes a B cell-specific membrane protein, was expressed extremely specifically in B cells, but the CD20 protein encoded by *MS4A1* was rarely detected, confirming the irreplaceability of the RNA modality. We also found that ADT could complement RNA-based clustering. For example, the simultaneous expression of T-cell markers (CD3 and CD4) was unexpectedly observed in two subclusters of B cells (B2 and B3) expressing canonical B-cell makers (CD19, CD20, and CD79; Fig. 2g). As this phenomenon could not be replicated using RNA data alone, this finding confirms the irreplaceability of the ADT modality.

Investigation of the uniqueness of chromatin accessibility in multi-omics integration at the ATAC level showed that ATAC contributed more than did ADT and RNA to the integration of a subcluster of $CD4^+$ naive T cells (Fig. 2h–j). We took the ratio of peak number of a cell to that of all cells as the representation of the cell accessibility level. RNA and ADT expression did not differ between these cells and their $CD4^+$ naive T-cell siblings, but surprisingly less accessibility level was observed at the ATAC layer (< 0.02, Supplementary Fig. 3). Gene ontology enrichment analysis [38] indicated that the inaccessible regions are related to T-cell activation, cell adhesion, and other immune functions. Therefore, we define this cluster as low chromatin accessible (LCA) naive $CD4^+$ T cells. Although this discovery needs to be verified further, it demonstrates the remarkable multi-omics integration capability of MIDAS.

2.3 MIDAS enables reliable trimodal mosaic integration

At present, trimodal sequencing techniques are still immature. Most of the existing datasets are unimodal or bimodal with various modality combinations. MIDAS is designed to integrate these diverse multimodal datasets, *i.e.* mosaic datasets. To evaluate the performance of MIDAS on mosaic integration, we further constructed 14 incomplete datasets based on the previously generated rectangular datasets including dogma-full and teadog-full datasets (Methods, Supplementary Table 1). Each mosaic dataset was generated by removing several modality-batch blocks from the full-modality dataset. Then, we took the rectangular integration results as the baseline, and examined whether MIDAS

could obtain comparable results on mosaic integration tasks. We assessed MIDAS's ability of batch correction, modality alignment, and biological conservation. Here we also focused on modality alignment because it guarantees accurate cross-modal inference for processes such as downstream imputation and knowledge transfer. For qualitative evaluation, we use UMAP to visualize the biological states and technical noises inferred from the individual and the joint input modalities (Fig. 3a, b, Supplementary Fig. 4, 5). Taking the dogma-paired-abc dataset for example, for each modality, the biological states were consistently distributed across different batches (Fig. 3a) whereas the technical noises were grouped by batches (Fig. 3b), indicating that the batch effect were well disentangled from the biological states. Similarly, the distributions of biological states and technical noise within batches were very similar across modalities (Fig. 3a, b), suggesting that MIDAS internally aligns different modalities in latent space. Moreover, the biological states of each cell type were grouped together and the cell type silhouettes were consistent across batches and modality combinations (Fig. 3a), reflecting robust conservation of the biological signals after mosaic integration.

To quantitatively evaluate MIDAS on mosaic integration, we proposed single-cell mosaic integration benchmarking (scMIB). scMIB extends scIB with modality alignment metrics, and defines each type of metrics on both observation space and latent space, resulting in 20 metrics in total (Methods, Supplementary Table 2). The obtained batch correction, modality alignment, biological conservation, and overall scores for paired+full, paired-abc, paired-ab, paired-ac, paired-bc, and diagonal+full tasks performed with the dogma and teadog datasets were similar to those obtained with rectangular integration (Fig. 3c, Supplementary Fig. 6a). MIDAS showed moderate performance in the dogma- and teadog-diagonal tasks, likely due to the lack of cell-to-cell correspondence across modalities in these tasks, which can be remedied via knowledge transfer (shown in Result 2.5).

scIB benchmarking showed that MIDAS outperformed methods that use the full-modality datasets (dogma- and teadog-full) when given incomplete datasets (paired+full, paired-abc, paired-ab, paired-ac, and paired-bc for dogma and teadog; Supplementary Fig. 6b, c). Even with the severely incomplete dogma- and teadog-diagonal+full datasets, the performance of MIDAS surpassed that of most the other methods.

The identification of cells' nearest neighbors based on individual dimensionality reduction results and comparison of neighborhood overlap among tasks showed that this overlap exceeded 0.75 for most tasks, except dogma-diagonal, when the number of neighbors reached 10,000 (Fig. 3d). As imputed omics data has been inferred to deteriorate the accuracy of gene regulatory inference in many cases [39], we evaluated the consistency of downstream analysis results obtained with the performance of different mosaic integration tasks with the dogma datasets. We validated the conservation of gene regulatory networks in the imputed data. In the dogma-paired+full task, for example, the regulatory network predicted from imputed data was consistent with that predicted from the ground-truth dogma-full data (Fig. 3e). These results indicate that the modality inference performed by MIDAS is reliable.

The manual annotation of cell types for the mosaic integration tasks and computation of their confusion matrices and micro-F1 scores with the dogma-full cell typing results serving as the ground truth showed that the cell type labels generated from the incomplete datasets, except dogma-diagonal, were largely consistent with the ground truth, with all micro-F1 scores exceeding 0.9 (Fig. 3f, Supplementary Fig. 7). The separation of monocytes and DCs was difficult in some mosaic experiments, mainly because the latter originate from the former [40] and likely also because the monocyte population in the dogma dataset was small.

2.4 MIDAS enables the atlas-level mosaic integration of trimodal PBMC data

We used MIDAS for the large-scale mosaic integration of 18 PBMC batches from bimodal sequencing platforms (*e.g.*, 10X Multiome, ASAP-seq, and CITE-seq) and the 9 batches from the DOGMA-seq and TEA-seq trimodal datasets (total, 27 batches from 10 platforms comprising 185,518 cells; Methods, Supplementary Table 3). Similar to the results obtained with the dogma-full and teadog-full datasets, MIDAS achieved satisfactory batch removal and biological conservation. UMAP visualization showed that the inferred biological states of different batches maintained a consistent PBMC population structure and conserved batch-specific (due mainly to differences in experimental design) biological information (Fig. 4a, Supplementary Fig. 8a). In addition, the technical noise was clearly grouped by batch (Supplementary Fig. 8b). These results suggest that the biological states and technical noise were disentangled well and the data could be used reliably in downstream analysis.

The manual labeling of cell types according to cluster markers achieved clearer separation and annotation than did automatic labeling by Seurat [13] (Fig. 4b). Consistent with the rectangular integration results (Fig. 2e), we identified all cell types known to be in the atlas, including B cells, conventional T-cell subsets, DP T cells, NK cells, unconventional T cells, and hematopoietic stem cells (HSCs), demonstrating the robustness of MIDAS. Remarkably, the integration of more datasets with MIDAS led to the identification of rare clusters and high-resolution cell typing. For example, whereas platelets could not be easily identified by dogma-full rectangular integration due to their extremely limited number (Fig. 2e), platelets from the DOGMA-seq dataset aggregated into a much larger cluster with recognizable

platelet markers in the PBMC atlas (Fig. 4c). In addition, the atlas contained more monocyte subclusters, including $CD14^+$, $CD16^+$, and $CD3^+CD14^+$ monocytes, than obtained with rectangular integration (Fig. 4d). Other cell types present in more subclusters in the atlas included $CD158e1^+$ NK cells, $CD4^+CD138^+CD202b^+$ T cells, and $RTKN2^+CD8^+$ T cells (Supplementary Fig. 9a).

Most batches in the atlas contained considerable numbers of LCA cells (Fig. 4e, Supplementary Fig. 9b) with $<$ 0.02 accessibility level, as did the DOGMA-seq dataset (Fig. 2i). The chromatin accessibility levels of cells in the atlas showed an obvious bimodal distribution, reflecting the existence of two ATAC patterns (Supplementary Fig. 9b). $CD8^+$ T-cell, $CD14^+$ monocyte, NK cell, B cell, and other clusters contained LCA cells (Fig. 4b, e) implying that LCA is common in various cell types.

2.5 MIDAS enables flexible and accurate knowledge transfer across modalities and datasets

To investigate the knowledge transfer capability of MIDAS, we re-partitioned the atlas dataset into reference (for atlas construction) and query (knowledge transfer target) datasets (Supplementary Table 3). By removing DOGMA-seq from the atlas, we obtained a reference dataset named atlas-no_dogma. To test the flexibility of knowledge transfer, we used DOGMA-seq to construct 14 query datasets: 1 rectangular integration and 7 incomplete mosaic datasets generated previously, and 6 rectangular integration datasets with fewer modalities (Methods, Supplementary Table 4). In consideration of real applications, we defined model and label knowledge transfer scenarios (Methods). In the model transfer scenario, knowledge was transferred implicitly through model parameters via transfer learning. In the label transfer scenario, knowledge was transferred explicitly through cell labels via reference mapping.

We assessed the performance of MIDAS in the model transfer scenario. For the transfer learned models, we used UMAP to visualize the inferred biological states and technical noise and scMIB and scIB for integration benchmarking, and compared the results of different tasks with those generated by *de novo* trained models. Transfer learning greatly improved performance on the dogma-diagonal, dogma-atac, dogma-rna, and dogma-paired-a tasks, with performance levels on the other tasks maintained (Fig. 5a–c, Supplementary Fig. 10, 11). For example, the *de novo* trained model failed to integrate well in the dogma-diagonal task due to lack of cell-to-cell correspondence across modalities (Fig. 5a), whereas the transfer learned model with atlas knowledge successfully aligned the biological states across batches and modalities and formed groups consistent with cell types (Fig. 5b). The results obtained by transfer learned models with all 14 datasets were not only comparable (Supplementary Fig. 11a), but also better than those obtained with many other methods, even with the complete dataset (Fig. 5c, Supplementary Fig. 11b).

To assess the performance of MIDAS in the label transfer scenario, we compared the widely used query-to-reference mapping [41, 42], reference-to-query mapping [14, 43], and our proposed reciprocal reference mapping (Methods). With each strategy, we aligned each query dataset to the reference dataset and transferred cell type labels through k-nearest neighbors, with the ground-truth cell type labels taken from the trimodal PBMC atlas. Visualization of the mapped biological states showed that reciprocal reference mapping with different query datasets yielded consistent results, with strong agreement with the atlas integration results obtained with the dogma-full dataset (Fig. 5d). Micro-F1 scores indicated that reciprocal reference mapping outperformed the query-to-reference and reference-to-query mapping strategies for various forms of query data, achieving robust and accurate label transfer and thereby avoiding the need for *de novo* downstream analysis (Fig. 5e).

Thus, MIDAS can be used to transfer atlas-level knowledge to various forms of users' datasets without expensive *de novo* training or complex downstream analysis.

3 Discussion

By modeling the single-cell mosaic data generative process, MIDAS can precisely disentangle biological states and technical noise from the input and robustly align modalities to support multi-source and heterogeneous integration analysis. Its performance in complete trimodal integration is superior to that of other methods, and it provides accurate and robust results when performing various mosaic integration tasks, which is beyond the scope of other tools. It also powerfully and reliably integrates large datasets, as demonstrated with the atlas-scale integration of publicly available PBMC multi-omics datasets. Moreover, MIDAS efficiently and flexibly transfers knowledge from reference to query datasets, enabling convenient handling of new multi-omics data. With superior performance in dimensionality reduction and batch correction, MIDAS supports accurate downstream biological analysis.

To our knowledge, MIDAS is the first model that supports simultaneous dimensionality reduction, modality complementing, and batch correction in single-cell trimodal mosaic integration. MIDAS accurately integrates mosaic data with missing modalities, achieving results comparable to the ground truth (rectangular integration) and superior to those obtained with the application of other methods to data with complete modalities. These distinct advantages

of MIDAS derive from the deep generative modeling, product of experts, information-theoretic disentanglement, and self-supervised modality alignment components of the algorithm, which are specifically designed and inherently competent for the heterogeneous integration of data with missing features and modalities. In addition, MIDAS is the first model that allows knowledge transfer across mosaic data modalities and batches in a highly flexible and reliable manner, enabling researchers to conquer the vast bodies of data produced with continuously emerging multi-omics techniques.

GLUE and uniPort, designed for the integration of unpaired single-cell trimodal data, align the global distributions of different modalities with the use of prior information to improve cell-wise correspondence. They may face challenges when little prior information about the inter-modality (*e.g.*, ATAC *vs.* ADT) correspondence of single cells is available. In addition, they are not designed to utilize paired data to enhance modality alignment. With the widespread adoption and rapid development of scMulti-omics sequencing technologies, paired data are rapidly becoming more common and will soon be ubiquitous. By leveraging paired cell data, MIDAS can learn to better align different modalities in a self-supervised manner and can thus play a more important role than other available methods in the blooming scMulti-omics era.

Recently proposed methods for scalable reference-to-query knowledge transfer for single-cell multimodal data have issues with generalization to unseen query data [41, 42] or the retention of information learned on the reference data [14, 43], which make the alignment of reference and query datasets difficult. In addition, they support limited numbers of specific modalities. The MIDAS knowledge transfer scheme stands out from these methods because it supports various types of mosaic query dataset and enables model transfer for sample-efficient mosaic integration and label transfer for automatic cell typing. Moreover, the generalization and retention problems are mitigated through a novel reciprocal reference mapping scheme.

We envision two major developmental directions for MIDAS. At present, MIDAS integrates only three modalities. By fine tuning the model architecture, we can achieve the integration of four or more modalities, overcoming the limitations of existing scMulti-omics sequencing technologies. In addition, the continuous incorporation of rapidly increasing bodies of newly generated scMulti-omics data is needed to update the model and improve the quality of the atlas. This process requires frequent model retraining, which is computationally expensive and time consuming. Thus, employing incremental learning [44] is an inevitable trend to achieve continuous mosaic integration without model retraining.

4 Methods

4.1 Deep generative modeling of mosaic single-cell multimodal data

For Cell $n \in \mathcal{N} = \{1, \ldots, N\}$ with batch ID $s_n \in \mathcal{S} = \{1, \ldots, S\}$, let $\boldsymbol{x}_n^m \in \mathbb{N}^{D_n^m}$ be the count vector of size D_n^m from Modality m, and $\boldsymbol{x}_n = \{\boldsymbol{x}_n^m\}_{m \in \mathcal{M}_n}$ the set of count vectors from the measured modalities $\mathcal{M}_n \subseteq \mathcal{M} = \{\text{ATAC}, \text{RNA}, \text{ADT}\}$. We define two modality-agnostic low-dimensional latent variables $\boldsymbol{c} \in \mathbb{R}^{D^c}$ and $\boldsymbol{u} \in \mathbb{R}^{D^u}$ to represent each cell's biological state and technical noise, respectively. To model the generative process of the observed variables $\boldsymbol{x}$ and s for each cell, we factorize the joint distribution of all variables as below:

$$\begin{aligned} p(\boldsymbol{x}, s, \boldsymbol{c}, \boldsymbol{u}) &= p(\boldsymbol{c})p(\boldsymbol{u})p(s|\boldsymbol{u})p(\boldsymbol{x}|\boldsymbol{c}, \boldsymbol{u}) \\ &= p(\boldsymbol{c})p(\boldsymbol{u})p(s|\boldsymbol{u}) \prod_{m \in \mathcal{M}_n} p(\boldsymbol{x}^m|\boldsymbol{c}, \boldsymbol{u}) \end{aligned} \tag{1}$$

where we assume that $\boldsymbol{c}$ and $\boldsymbol{u}$ are independent of each other and the batch ID s only depends on $\boldsymbol{u}$ in order to facilitate the disentanglement of both latent variables, and that the count variables $\{\boldsymbol{x}^m\}_{m \in \mathcal{M}_n}$ from different modalities are conditional independent given $\boldsymbol{c}$ and $\boldsymbol{u}$.

Based on the above factorization, we define a generative model for $\boldsymbol{x}$ and s as follows:

$$p(\boldsymbol{c}) = \text{Normal}(\boldsymbol{c}|\boldsymbol{0}, \boldsymbol{I}) \tag{2}$$

$$p(\boldsymbol{u}) = \text{Normal}(\boldsymbol{u}|\boldsymbol{0}, \boldsymbol{I}) \tag{3}$$

$$\boldsymbol{\pi} = g^s\left(\boldsymbol{u}; \boldsymbol{\theta}^s\right) \tag{4}$$

$$p_{\boldsymbol{\theta}}(s|\boldsymbol{u}) = \text{Categorical}(s|\boldsymbol{\pi}) \tag{5}$$

$$\boldsymbol{\lambda}^m = g^m\left(\boldsymbol{c}, \boldsymbol{u}; \boldsymbol{\theta}^m\right) \quad \text{for } m \in \mathcal{M}_n \tag{6}$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^m|\boldsymbol{c}, \boldsymbol{u}) = \begin{cases} \text{Bernoulli}(\boldsymbol{x}^m|\boldsymbol{\lambda}^m) & \text{if } m = \text{ATAC} \\ \text{Poisson}(\boldsymbol{x}^m|\boldsymbol{\lambda}^m) & \text{if } m \in \{\text{RNA}, \text{ADT}\} \end{cases} \quad \text{for } m \in \mathcal{M}_n \tag{7}$$

where the priors $p(\boldsymbol{c})$ and $p(\boldsymbol{u})$ are set as standard normals. The likelihood $p_{\boldsymbol{\theta}}(s|\boldsymbol{u})$ is set as a categorical distribution with probability vector $\boldsymbol{\pi} \in \Delta^{S-1}$ generated through a batch-ID decoder g^s which is a neural network with learnable parameters $\boldsymbol{\theta}^s$. The likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}^m|\boldsymbol{c},\boldsymbol{u})$ is set as a Bernoulli distribution with mean $\boldsymbol{\lambda}^m \in [0,1]^{D_n^m}$ when $m = \text{ATAC}$, and as a Poisson distribution with mean $\boldsymbol{\lambda}^m \in \mathbb{R}_+^{D_n^m}$ when $m \in \{\text{RNA}, \text{ADT}\}$, where $\boldsymbol{\lambda}^m$ is generated through a modality decoder neural network g^m parameterized by $\boldsymbol{\theta}^m$. To mitigate overfitting and improve generalization, we share parameters of the first few layers of different modality decoders $\{g^m\}_{m\in\mathcal{M}}$ (the gray parts of the decoders in Fig. 1b middle). The corresponding graphical model is shown in Fig. 1b (left).

Given the observed data $\{\boldsymbol{x}_n, s_n\}_{n\in\mathcal{N}}$, we aim to fit the model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}^s, \{\boldsymbol{\theta}^m\}_{m\in\mathcal{M}}\}$ and meanwhile infer the posteriors of latent variables $\{\boldsymbol{c}, \boldsymbol{u}\}$ for each cell. This can be achieved by using the SGVB [33] which maximizes the expected Evidence Lower Bound (ELBO) for individual datapoints. The ELBO for each individual datapoint $\{\boldsymbol{x}_n, s_n\}$ can be written as:

$$\begin{aligned}
\text{ELBO}(\boldsymbol{\theta},\boldsymbol{\phi};\boldsymbol{x}_n,s_n) &\triangleq \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_n,s_n,\boldsymbol{c},\boldsymbol{u})}{q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)}\right] \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n,s_n|\boldsymbol{c},\boldsymbol{u})\right] - \text{KL}\left(q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)\|p(\boldsymbol{c},\boldsymbol{u})\right) \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)}\left[\log p_{\boldsymbol{\theta}}(s_n|\boldsymbol{u}) + \sum_{m\in\mathcal{M}_n} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}_n^m|\boldsymbol{c},\boldsymbol{u})\right] - \text{KL}\left(q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)\|p(\boldsymbol{c},\boldsymbol{u})\right)
\end{aligned} \tag{8}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)$, with learnable parameters $\boldsymbol{\phi}$, is the variational approximation of the true posterior $p(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)$ and is typically implemented by neural networks.

4.2 Scalable variational inference via the Product of Experts

Let $M = |\mathcal{M}|$ be the total modality number, since there are $(2^M - 1)$ possible modality combinations for the count data $\boldsymbol{x}_n = \{\boldsymbol{x}_n^m\}_{m\in\mathcal{M}_n\subseteq\mathcal{M}}$, naively implementing $q_{\boldsymbol{\phi}}(\boldsymbol{c},\boldsymbol{u}|\boldsymbol{x}_n,s_n)$ in equation (8) requires $(2^M - 1)$ different neural networks to handle different cases of input $(\boldsymbol{x}_n, s_n)$, making inference unscalable. Let $\boldsymbol{z} = \{\boldsymbol{c},\boldsymbol{u}\}$, inspired by [45] which utilizes the Product of Experts (PoEs) to implement variational inference in a combinatorial way, we factorize the posterior $p(\boldsymbol{z}|\boldsymbol{x}_n,s_n)$ and define its variational approximation $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n,s_n)$ as follows:

$$\begin{aligned}
p(\boldsymbol{z}|\boldsymbol{x}_n,s_n) &= \frac{p(\boldsymbol{z})p(s_n|\boldsymbol{z})p(\boldsymbol{x}_n|\boldsymbol{z})}{p(\boldsymbol{x}_n,s_n)} \\
&= \frac{p(\boldsymbol{z})}{p(\boldsymbol{x}_n,s_n)} p(s_n|\boldsymbol{z}) \prod_{m\in\mathcal{M}_n} p(\boldsymbol{x}_n^m|\boldsymbol{z}) \\
&= \frac{p(\boldsymbol{z})}{p(\boldsymbol{x}_n,s_n)} \frac{p(s_n)p(\boldsymbol{z}|s_n)}{p(\boldsymbol{z})} \prod_{m\in\mathcal{M}_n} \frac{p(\boldsymbol{x}_n^m)p(\boldsymbol{z}|\boldsymbol{x}_n^m)}{p(\boldsymbol{z})} \\
&= \frac{p(s_n)}{p(\boldsymbol{x}_n,s_n)} \left(\prod_{m\in\mathcal{M}_n} p(\boldsymbol{x}_n^m)\right) p(\boldsymbol{z}) \frac{p(\boldsymbol{z}|s_n)}{p(\boldsymbol{z})} \prod_{m\in\mathcal{M}_n} \frac{p(\boldsymbol{z}|\boldsymbol{x}_n^m)}{p(\boldsymbol{z})} \\
&\approx \frac{p(s_n)}{p(\boldsymbol{x}_n,s_n)} \left(\prod_{m\in\mathcal{M}_n} p(\boldsymbol{x}_n^m)\right) p(\boldsymbol{z}) \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)}{p(\boldsymbol{z})} \prod_{m\in\mathcal{M}_n} \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)}{p(\boldsymbol{z})} \\
&\triangleq q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n,s_n)
\end{aligned} \tag{9}$$

where $q_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)$ are the variational approximations of the true posteriors $p(\boldsymbol{z}|s_n)$ and $p(\boldsymbol{z}|\boldsymbol{x}_n^m)$, respectively. From equation (9) we further get:

$$\begin{aligned}
q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n,s_n) &\propto p(\boldsymbol{z}) \underbrace{\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)}{p(\boldsymbol{z})}}_{\triangleq C^s\cdot\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)} \prod_{m\in\mathcal{M}_n} \underbrace{\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)}{p(\boldsymbol{z})}}_{\triangleq C^m\cdot\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)} \\
&\propto p(\boldsymbol{z})\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n) \prod_{m\in\mathcal{M}_n} \widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)
\end{aligned} \tag{10}$$

where we set $q_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)$ to be Gaussians. We define $\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n) = \frac{1}{C^s}\cdot\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)}{p(\boldsymbol{z})}$ and $\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m) = \frac{1}{C^m}\cdot\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)}{p(\boldsymbol{z})}$ to be the normalized quotients of Gaussians with normalizing constants C^s and C^m, respectively.

Since the quotient of Gaussians is an unnormalized Gaussian, $\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)$ and $\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)$ are also Gaussians. Thus, in equation (10) the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n, s_n)$ is proportional to the product of individual Gaussians (or "experts"), indicating that itself is a Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Lambda}$:

$$\begin{aligned}\boldsymbol{\mu} &= \left(\sum_i \boldsymbol{\mu}_i \boldsymbol{\Lambda}_i^{-1}\right)\left(\sum_i \boldsymbol{\Lambda}_i^{-1}\right)^{-1} \\ \boldsymbol{\Lambda} &= \left(\sum_i \boldsymbol{\Lambda}_i^{-1}\right)^{-1}\end{aligned} \tag{11}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Lambda}_i$ are respectively the mean and covariance of the i-th individual Gaussian. We further assume $\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m)$ and $\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)$ are isotropic Gaussians and define them as follows:

$$(\boldsymbol{\mu}^m, \boldsymbol{\nu}^m) = f^m(\boldsymbol{x}_n^m; \boldsymbol{\phi}^m) \tag{12}$$
$$\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m) = \text{Normal}\left(\boldsymbol{z}|\boldsymbol{\mu}^m, \boldsymbol{\nu}^m \boldsymbol{I}\right) \tag{13}$$
$$(\boldsymbol{\mu}^s, \boldsymbol{\nu}^s) = f^s(s_n; \boldsymbol{\phi}^s) \tag{14}$$
$$\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n) = \text{Normal}(\boldsymbol{z}|\boldsymbol{\mu}^s, \boldsymbol{\nu}^s \boldsymbol{I}) \tag{15}$$

where f^m is the modality encoder parameterized by $\boldsymbol{\phi}^m$, and f^s the batch-ID encoder parameterized by $\boldsymbol{\phi}^s$, both of which are neural networks for generating Gaussian mean-variance pairs. In doing this, $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n, s_n)$ is modularized into $(M+1)$ neural networks to handle (2^M-1) different modality combinations, increasing the model's scalability. Similar to the modality decoders, we also share parameters of the last few layers of different modality encoders $\{f^m\}_{m\in\mathcal{M}}$ (the gray parts of the encoders in Fig. 1b middle) to improve generalization.

4.3 Handling missing features via padding and masking

For each modality, as different batches can have different feature sets (*e.g.*, genes for RNA modality), it is hard to use a fixed-size neural network to handle these batches. To remedy this, we first convert $\boldsymbol{x}_n^m$ of variable size into a fixed size vector for inference. For Modality m, let $\mathcal{F}_s^m$ be the features of batch s, and $\mathcal{F}^m = \bigcup_{s\in\mathcal{S}} \mathcal{F}_s^m$ the feature union of all batches. The missing features of batch s can then be defined as $\overline{\mathcal{F}}_s^m = \mathcal{F}^m \setminus \mathcal{F}_s^m$. We pad $\boldsymbol{x}_n^m$ of size D_n^m with zeros corresponding to its missing features $\overline{\mathcal{F}}_{s_n}^m$ through a zero-padding function h:

$$\widetilde{\boldsymbol{x}}_n^m = h(\boldsymbol{x}_n^m) \tag{16}$$

where $\widetilde{\boldsymbol{x}}_n^m$ is the zero-padded count vector of constant size $D^m = |\mathcal{F}^m|$. The modality encoding process is thus decomposed as:

$$\begin{aligned}(\boldsymbol{\mu}^m, \boldsymbol{\nu}^m) &= f^m(\boldsymbol{x}_n^m; \boldsymbol{\phi}^m) \\ &= \widehat{f}^m\left(h(\boldsymbol{x}_n^m); \boldsymbol{\phi}^m\right) \\ &= \widehat{f}^m\left(\widetilde{\boldsymbol{x}}_n^m; \boldsymbol{\phi}^m\right)\end{aligned} \tag{17}$$

where $\widehat{f}^m$ is the latter part of the modality encoder to handle a fixed size input $\widetilde{\boldsymbol{x}}_n^m$. On the other hand, to calculate the likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{x}_n^m|\boldsymbol{c}_n, \boldsymbol{u}_n)$ we also need to generate a mean $\boldsymbol{\lambda}_n^m$ of variable size for $\boldsymbol{x}_n^m$. To achieve this, we decompose the modality decoding process as follows:

$$\begin{aligned}\boldsymbol{\lambda}_n^m &= g^m\left(\boldsymbol{c}_n, \boldsymbol{u}_n; \boldsymbol{\theta}^m\right) \\ &= h^{-1}\left(\widehat{g}^m\left(\boldsymbol{c}_n, \boldsymbol{u}_n; \boldsymbol{\theta}^m\right)\right) \\ &= h^{-1}\left(\widetilde{\boldsymbol{\lambda}}_n^m\right)\end{aligned} \tag{18}$$

where $\widehat{g}^m$ is the front part of the modality decoder to generate the mean $\widetilde{\boldsymbol{\lambda}}_n^m$ of fixed size D^m, and h^{-1}, the inverse function of h, is the mask function to remove the padded missing features $\overline{\mathcal{F}}_s^m$ from $\widetilde{\boldsymbol{\lambda}}_n^m$ to generate $\boldsymbol{\lambda}_n^m$. Note that $\widetilde{\boldsymbol{\lambda}}_n^m$ can also be taken as the imputed values for downstream analyses (Methods 4.8 and 4.9).

4.4 Self-supervised modality alignment

To achieve cross-modal inference in downstream tasks, we resort to aligning different modalities in the latent space. Leveraging self-supervised learning, we first use each cell's multimodal observation $\{\{\boldsymbol{x}_n^m\}_{m\in\mathcal{M}_n}, s_n\}$ to construct

unimodal observations $\{\boldsymbol{x}_n^m, s_n\}_{m\in\mathcal{M}_n}$, each of which is associated with the latent variables $\boldsymbol{z}^m = \{\boldsymbol{c}^m, \boldsymbol{u}^m\}$. Then, we construct a pretext task, which enforces modality alignment by regularizing on the joint space of unimodal varialtional posteriors with the dispersion of latent variables as a penalty (Fig. 1 upper right), corresponding to a modality alignment loss:

$$\begin{aligned} l^{\text{MOD}}(\boldsymbol{\phi}; \boldsymbol{x}_n, s_n) &\triangleq \int v(\widetilde{\boldsymbol{z}}) q_{\boldsymbol{\phi}}(\widetilde{\boldsymbol{z}}|\boldsymbol{x}_n, s_n) d\widetilde{\boldsymbol{z}} \\ &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\widetilde{\boldsymbol{z}}|\boldsymbol{x}_n, s_n)} v(\widetilde{\boldsymbol{z}}) \end{aligned} \tag{19}$$

where $\widetilde{\boldsymbol{z}} = \{\boldsymbol{z}^m\}_{m\in\mathcal{M}_n}$ is the set of latent variables, and $q_{\boldsymbol{\phi}}(\widetilde{\boldsymbol{z}}|\boldsymbol{x}_n, s_n)$ represents the joint distribution of unimodal variational posteriors since:

$$\begin{aligned} q_{\boldsymbol{\phi}}(\widetilde{\boldsymbol{z}}|\boldsymbol{x}_n, s_n) &= q_{\boldsymbol{\phi}}(\{\boldsymbol{z}^m\}_{m\in\mathcal{M}_n}|\boldsymbol{x}_n, s_n) \\ &= \prod_{m\in\mathcal{M}_n} q_{\boldsymbol{\phi}}(\boldsymbol{z}^m|\boldsymbol{x}_n, s_n) \\ &= \prod_{m\in\mathcal{M}_n} q_{\boldsymbol{\phi}}(\boldsymbol{z}^m|\boldsymbol{x}_n^m, s_n) \end{aligned} \tag{20}$$

In equation (19), $v(\widetilde{\boldsymbol{z}})$ is the Mean Absolute Deviation, which measures the dispersion among different elements in $\widetilde{\boldsymbol{z}}$ and is used to regularize $q_{\boldsymbol{\phi}}(\widetilde{\boldsymbol{z}}|\boldsymbol{x}_n, s_n)$. It is defined as:

$$v(\widetilde{\boldsymbol{z}}) \triangleq \frac{1}{|\mathcal{M}_n|} \sum_{m\in\mathcal{M}_n} \|\boldsymbol{z}^m - \bar{\boldsymbol{z}}\|_2 \tag{21}$$

where $\bar{\boldsymbol{z}} = \frac{1}{|\mathcal{M}_n|}\sum_{m\in\mathcal{M}_n} \boldsymbol{z}^m$ is the mean and $\|\cdot\|_2$ the Euclidean distance.

Note that the compuation of $q_{\boldsymbol{\phi}}(\boldsymbol{z}^m|\boldsymbol{x}_n^m, s_n)$ in equation (20) is efficient. Since $q_{\boldsymbol{\phi}}(\boldsymbol{z}^m|\boldsymbol{x}_n^m, s_n) = q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m, s_n)|_{\boldsymbol{z}=\boldsymbol{z}^m}$, according to equation (10) we have:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m, s_n) \propto p(\boldsymbol{z})\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|s_n)\widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m) \tag{22}$$

As the mean and covariance of each Gaussian term on the right-hand side of equation (22) was already obtained when inferring $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n, s_n)$ (equation (10)), the mean and covariance of $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n^m, s_n)$ can be directly calculated using equation (11), avoiding the need of passing each constructed unimodal observation to the encoders.

4.5 Information-theoretic disentanglement of latent variables

To better disentangle the biological state $\boldsymbol{c}$ and the technical noise $\boldsymbol{u}$, we adopt an information-theoretic approach, the Information Bottleneck (IB) [26], to control the information flow during inference. We define two types of IB, where the technical IB prevents batch-specific information being encoded into $\boldsymbol{c}$ by minimizing the Mutual Information (MI) between s and $\boldsymbol{c}$, and the biological IB prevents biological information being encoded into $\boldsymbol{u}$ by minimizing the MI between $\boldsymbol{x}$ and $\boldsymbol{u}$ (Fig. 1 bottom right). Let $\mathrm{I}(\cdot,\cdot)$ denote the MI between two variables, considering both $\mathrm{I}(s, \boldsymbol{c})$ and

$\mathrm{I}(\boldsymbol{x}, \boldsymbol{u})$ we have:

$$
\begin{aligned}
\mathrm{I}(s, \boldsymbol{c}) + \mathrm{I}(\boldsymbol{x}, \boldsymbol{u}) &= \mathbb{E}_{p(s,\boldsymbol{c})}\left[\log \frac{p(s,\boldsymbol{c})}{p(s)p(\boldsymbol{c})}\right] + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{u})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{u})}{p(\boldsymbol{x})p(\boldsymbol{u})}\right] \\
&= \mathbb{E}_{p(s,\boldsymbol{c})}\Big[\log \underbrace{p(s|\boldsymbol{c})}_{\approx p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})}\Big] \underbrace{-\mathbb{E}_{p(s,\boldsymbol{c})}\left[\log p(s)\right]}_{\text{const.}} + \mathbb{E}_{p(\boldsymbol{x},\boldsymbol{u})}\left[\log \frac{p(\boldsymbol{u}|\boldsymbol{x})}{p(\boldsymbol{u})}\right] \\
&\approx \mathbb{E}_{p(\boldsymbol{x},s)}\left[\mathbb{E}_{p(\boldsymbol{c}|\boldsymbol{x},s)}\left[\log p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})\right]\right] + \text{const.} + \mathbb{E}_{p(\boldsymbol{x},s)}\left[\mathbb{E}_{p(\boldsymbol{u}|\boldsymbol{x},s)}\left[\log \frac{p(\boldsymbol{u}|\boldsymbol{x})}{p(\boldsymbol{u})}\right]\right] \\
&= \mathbb{E}_{p(\boldsymbol{x},s)}\left[\mathbb{E}_{p(\boldsymbol{c}|\boldsymbol{x},s)}\left[\log p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})\right] + \mathbb{E}_{p(\boldsymbol{u}|\boldsymbol{x},s)}\left[\log \frac{p(\boldsymbol{u}|\boldsymbol{x})}{p(\boldsymbol{u})}\right]\right] + \text{const.} \\
&\approx \frac{1}{N}\sum_n \left[\mathbb{E}_{p(\boldsymbol{c}|\boldsymbol{x}_n,s_n)}\left[\log p_{\widehat{\boldsymbol{\alpha}}}(s_n|\boldsymbol{c})\right] + \mathbb{E}_{p(\boldsymbol{u}|\boldsymbol{x}_n,s_n)}\left[\log \frac{p(\boldsymbol{u}|\boldsymbol{x}_n)}{p(\boldsymbol{u})}\right]\right] + \text{const.} \\
&\approx \frac{1}{N}\sum_n \Bigg[\underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_n,s_n)}\left[\log p_{\widehat{\boldsymbol{\alpha}}}(s_n|\boldsymbol{c})\right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{u}|\boldsymbol{x}_n,s_n)}\left[\log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{u}|\boldsymbol{x}_n)}{p(\boldsymbol{u})}\right]}_{\triangleq \widehat{l}^{\,\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n,s_n,\widehat{\boldsymbol{\alpha}})}\Bigg] + \text{const.} \\
&= \frac{1}{N}\sum_n \left[\widehat{l}^{\,\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n,s_n,\widehat{\boldsymbol{\alpha}})\right] + \text{const.}
\end{aligned}
\tag{23}
$$

where $p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})$ is a learned likelihood with parameters $\widehat{\boldsymbol{\alpha}}$, and $\frac{q_{\boldsymbol{\phi}}(\boldsymbol{u}|\boldsymbol{x}_n)}{p(\boldsymbol{u})} \propto \prod_{m\in\mathcal{M}_n} \widetilde{q}_{\boldsymbol{\phi}}(\boldsymbol{u}|\boldsymbol{x}_n^m)$ can be computed via equation (11). From equation (23), minimizing $(\mathrm{I}(s,\boldsymbol{c}) + \mathrm{I}(\boldsymbol{x},\boldsymbol{u}))$ approximately equals to minimizing $\widehat{l}^{\,\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n,s_n,\widehat{\boldsymbol{\alpha}})$ w.r.t $\boldsymbol{\phi}$ for all cells. For $p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})$, we model it as:

$$
p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c}) = \mathrm{Categorical}\left(s|r(\boldsymbol{c};\widehat{\boldsymbol{\alpha}})\right) \tag{24}
$$

where r is a classifier neural network parameterized by $\widehat{\boldsymbol{\alpha}}$. To learn the classifier, we minimize the following expected negative log-likelihood w.r.t $\widehat{\boldsymbol{\alpha}}$:

$$
\begin{aligned}
&\mathbb{E}_{p(\boldsymbol{c},s)}\left[-\log p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})\right] \\
=&\mathbb{E}_{p(\boldsymbol{x},s)}\mathbb{E}_{p(\boldsymbol{c}|\boldsymbol{x},s)}\left[-\log p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})\right] \\
\approx&\frac{1}{N}\sum_n \mathbb{E}_{p(\boldsymbol{c}|\boldsymbol{x}_n,s_n)}\left[-\log p_{\widehat{\boldsymbol{\alpha}}}(s_n|\boldsymbol{c})\right] \\
\approx&\frac{1}{N}\sum_n \underbrace{\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_n,s_n)}\left[-\log p_{\widehat{\boldsymbol{\alpha}}}(s_n|\boldsymbol{c})\right]}_{\triangleq \widehat{l}^{\,r}(\widehat{\boldsymbol{\alpha}};\boldsymbol{x}_n,s_n,\boldsymbol{\phi})} \\
=&\frac{1}{N}\sum_n \widehat{l}^{\,r}(\widehat{\boldsymbol{\alpha}};\boldsymbol{x}_n,s_n,\boldsymbol{\phi})
\end{aligned}
\tag{25}
$$

Thus, minimizing $\mathbb{E}_{p(\boldsymbol{c},s)}\left[-\log p_{\widehat{\boldsymbol{\alpha}}}(s|\boldsymbol{c})\right]$ approximately equals to minimizing $\widehat{l}^{\,r}(\widehat{\boldsymbol{\alpha}};\boldsymbol{x}_n,s_n,\boldsymbol{\phi})$ w.r.t $\widehat{\boldsymbol{\alpha}}$ for all cells.

To further enhance latent disentanglement for cross-modal inference, for each modality m, we also minimize $(\mathrm{I}(s,\boldsymbol{c}^m) + \mathrm{I}(\boldsymbol{x}^m,\boldsymbol{u}^m))$. Similar to equation (23), this can be achieved by minimizing $\widehat{l}^{\,\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n^m,s_n,\boldsymbol{\alpha}^m)$ where $\boldsymbol{\alpha}^m$ is the parameters of the classifier neural network r^m to generate the likelihood $p_{\boldsymbol{\alpha}^m}(s|\boldsymbol{c}^m) = \mathrm{Categorical}\left(r^m(\boldsymbol{c}^m;\boldsymbol{\alpha}^m)\right)$. Together with $\widehat{l}^{\,\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n,s_n,\widehat{\boldsymbol{\alpha}})$ defined in equation (23), the total IB loss is defined as:

$$
l^{\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n,s_n,\boldsymbol{\alpha}) \triangleq \widehat{l}^{\,\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n,s_n,\widehat{\boldsymbol{\alpha}}) + \sum_{m\in\mathcal{M}_n} \widehat{l}^{\,\mathrm{IB}}(\boldsymbol{\phi};\boldsymbol{x}_n^m,s_n,\boldsymbol{\alpha}^m) \tag{26}
$$

where $\boldsymbol{\alpha} = \{\widehat{\boldsymbol{\alpha}}, \{\boldsymbol{\alpha}^m\}_{m\in\mathcal{M}}\}$. To learn $p_{\boldsymbol{\alpha}^m}(s|\boldsymbol{c}^m)$, we can also minimize $\mathbb{E}_{p(\boldsymbol{c}^m,s)}\left[-\log p_{\boldsymbol{\alpha}^m}(s|\boldsymbol{c}^m)\right]$, which corresponds to minimizing $\widehat{l}^{\,r}(\boldsymbol{\alpha}^m;\boldsymbol{x}_n^m,s_n,\boldsymbol{\phi})$ according to equation (25). Considering the classifier loss $\widehat{l}^{\,r}(\widehat{\boldsymbol{\alpha}};\boldsymbol{x}_n,s_n,\boldsymbol{\phi})$ defined in equation (25), the total classifier loss is defined as:

$$
l^{r}(\boldsymbol{\alpha};\boldsymbol{x}_n,s_n,\boldsymbol{\phi}) = \widehat{l}^{\,r}(\widehat{\boldsymbol{\alpha}};\boldsymbol{x}_n,s_n,\boldsymbol{\phi}) + \sum_{m\in\mathcal{M}_n} \widehat{l}^{\,r}(\boldsymbol{\alpha}^m;\boldsymbol{x}_n^m,s_n,\boldsymbol{\phi}) \tag{27}
$$

4.6 Training MIDAS

To train the encoders and decoders of MIDAS, considering the training objectives defined in equations (8), (19), and (26), we minimize the following objective w.r.t $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ for all observations $\{\boldsymbol{x}_n, s_n\}_{n\in\mathcal{N}}$:

$$l^{f,g}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_n, s_n, \boldsymbol{\alpha}) = l^{\text{ELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_n, s_n) + l^{\text{MOD}}(\boldsymbol{\phi}; \boldsymbol{x}_n, s_n) + l^{\text{IB}}(\boldsymbol{\phi}; \boldsymbol{x}_n, s_n, \boldsymbol{\alpha}) \tag{28}$$

where $l^{\text{ELBO}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_n, s_n) \triangleq -\text{ELBO}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_n, s_n)$. Since $\boldsymbol{\alpha}$ is unknown and the learning of $\boldsymbol{\alpha}$ depends on $\boldsymbol{\phi}$ as in equation (27), we iteratively minimize equations (28) and (27) with Stochastic Gradient Descent (SGD), forming the MIDAS training algorithm (Algorithm 1).

Algorithm 1 The MIDAS training algorithm.

Input: A single-cell multimodal mosaic dataset $\{\boldsymbol{x}_n, s_n\}_{n\in\mathcal{N}}$
Output: Encoder parameters $\boldsymbol{\theta}$, decoder parameters $\boldsymbol{\phi}$, and classifier parameters $\boldsymbol{\alpha}$

1: Randomly initialize parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\alpha}\}$
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Sample a minibatch $\{\boldsymbol{x}_n, s_n\}_{n\in\mathcal{N}_t}$ from the dataset, where $\mathcal{N}_t \subset \mathcal{N}$
4: Freeze $\boldsymbol{\phi}$ and update $\boldsymbol{\alpha}$ via SGD with loss $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} l^r(\boldsymbol{\alpha}; \boldsymbol{x}_n, s_n, \boldsymbol{\phi})$ ▷ See equation (27)
5: Freeze $\boldsymbol{\alpha}$ and update $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ via SGD with loss $\frac{1}{|\mathcal{N}_t|}\sum_{n\in\mathcal{N}_t} l^{f,g}(\boldsymbol{\theta}, \boldsymbol{\phi}; \boldsymbol{x}_n, s_n, \boldsymbol{\alpha})$ ▷ See equation (28)
6: **end for**

4.7 Mosaic integration on latent space

A key goal of single-cell mosaic integration is to extract biological meaningful low-dimensional representations from the mosaic data for downstream analysis. To achieve this, for each cell we first use the trained MIDAS to infer the mean and variance of the latent posterior $q_{\boldsymbol{\phi}}(\boldsymbol{c}, \boldsymbol{u}|\boldsymbol{x}_n, s_n)$ through equation (10), and then take the Maximum A Posterior (MAP) estimation of the biological state $\boldsymbol{c}$ as the cell's low-dimensional representation. Since $q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_n, s_n)$ is Gaussian, the MAP estimation of $\boldsymbol{c}$ corresponds to the inferred mean of $q_{\boldsymbol{\phi}}(\boldsymbol{c}|\boldsymbol{x}_n, s_n)$.

4.8 Imputation for missing modalities and features

Based on the latent posterior $q_{\boldsymbol{\phi}}(\boldsymbol{c}, \boldsymbol{u}|\boldsymbol{x}_n, s_n)$ inferred from the single-cell mosaic data (Methods 4.7), it is straightforward to impute missing modalities and features. We first pass the inferred posterior means of $\boldsymbol{c}$ and $\boldsymbol{u}$ to the decoders to generate padded feature mean $\widetilde{\boldsymbol{\lambda}}_n^m$ for each modality $m \in \mathcal{M}$ via equation (18). Then, we sample from a Bernoulli distribution with mean $\widetilde{\boldsymbol{\lambda}}_n^{\text{ATAC}}$ to generate the imputed ATAC counts, and from two Poisson distributions with means $\widetilde{\boldsymbol{\lambda}}_n^{\text{RNA}}$ and $\widetilde{\boldsymbol{\lambda}}_n^{\text{ADT}}$ to generate the imputed RNA and ADT counts, respectively.

4.9 Batch correction via latent variable manipulation

Besides performing mosaic integration on the latent space (Methods 4.7), we can also perform it on the feature space, *i.e.*, imputing missing values and correcting batch effect for the count data. Mosaic integration on feature space is important since it is required by many downstream tasks such as differential expression analysis and cell typing.

Based on the inferred posterior means of latent variables $\boldsymbol{c}$ and $\boldsymbol{u}$ (Methods 4.8) of each cell, we can perform imputation and batch correction simultaneously by manipulating the technical noise $\boldsymbol{u}$. Concretely, let $\boldsymbol{c}_n$ and $\boldsymbol{u}_n$ be the posterior means of $\boldsymbol{c}$ and $\boldsymbol{u}$ for Cell n, respectively, we first calculate the mean of $\boldsymbol{u}_n$ within each batch s:

$$\bar{\boldsymbol{u}}_s = \frac{1}{|\mathcal{N}_s|}\sum_{n\in\mathcal{N}_s} \boldsymbol{u}_n \tag{29}$$

where $\mathcal{N}_s \subseteq \mathcal{N}$ is the set of cell-IDs belonging to batch s. Next, we calculate the mean of $\bar{\boldsymbol{u}}_s$ over all batches:

$$\bar{\boldsymbol{u}} = \frac{1}{S}\sum_s \bar{\boldsymbol{u}}_s \tag{30}$$

Then, we find a mean $\bar{\boldsymbol{u}}_{S^*}$ which is the closest to $\bar{\boldsymbol{u}}$, and treat $\bar{\boldsymbol{u}}_{S^*}$ as a "standard" technical noise, where:

$$S^* = \arg\min_s \|\bar{\boldsymbol{u}}_s - \bar{\boldsymbol{u}}\|_2 \tag{31}$$

Finally, for each cell we correct the batch effect by substituting $\boldsymbol{u}_n$ with a common $\bar{\boldsymbol{u}}_{S^*}$, and pass $\{\boldsymbol{c}_n, \bar{\boldsymbol{u}}_{S^*}\}$ to the decoders to generate imputed and batch-corrected data (similar to Methods 4.8 but here we use $\{\boldsymbol{c}_n, \bar{\boldsymbol{u}}_{S^*}\}$ instead of $\{\boldsymbol{c}_n, \boldsymbol{u}_n\}$ to correct batch effect).

4.10 Model transfer via transfer learning

When MIDAS has been pre-trained on a reference dataset, we can conduct model transfer to transfer the model's learned knowledge to a query dataset through transfer learning, *i.e.*, on the query dataset we fine-tune the pre-trained model instead of train the model from scratch. Since compared to the reference dataset, the query dataset can contain different number of batches collected from different platforms, the batch-ID related modules need to be redefined. Thus, during transfer learning, we reparameterize and reinitialize the batch-ID encoder and decoder $\{f^s, g^s\}$ and the batch classifiers $\{r, \{r^m\}_{m\in\mathcal{M}}\}$, and only fine-tune the modality encoders and decoders $\{f^m, g^m\}_{m\in\mathcal{M}}$.

A core advantage of our model transfer scheme is that it can flexibly transfer the knowledge of multimodal data to various types of query datasets, even to those with fewer modalities, improving the *de novo* integration of single-cell data.

4.11 Label transfer via reciprocal reference mapping

Whilst model transfer implicitly transfers knowledge through model parameters, label transfer explicitly transfers knowledge in the form of data labels. These labels can be different kinds of downstream analysis results such as cell types, cell cycles, or pseudotime. Through accurate label transfer, we can not only avoid the expensive *de novo* downstream analysis, but also improve the label quality.

Typically, the first step of label transfer is reference mapping, which aligns the query cells with the reference cells so that labels can be transferred reliably. For MIDAS, we can naively achieve reference mapping in two ways: (1) mapping the query data onto the reference space, *i.e.*, applying the model pre-trained on the reference data to infer the biological states for the query data [41, 42], and (2) mapping the reference data onto the query space, *i.e.*, applying the model transfer-learned on the query data (Methods 4.10) to infer the biological states for the reference data [14, 43]. However, the first way suffers from the 'generalization problem' since the pre-trained model is hard to generalize to the query data which usually contains unseen technical variations, while the second way suffers from the 'forgetting problem' since the transfer-learned model may lose information learned on the reference data, affecting the inferred biological states.

To tackle both problems, we propose an reciprocal reference mapping scheme, where we fine-tune the pre-trained model on the query dataset to avoid the generalization problem, and meanwhile feed the model with the past data sampled from the reference dataset to prevent forgetting. In doing this, the model can find a mapping suitable for both reference and query datasets. We can then align the two datasets by inferring the biological states, through which the reference labels can be transferred to the query data based on nearest neighbors. Similar to model transfer (Methods 4.10), in label transfer knowledge can also be flexibly and accurately transferred to various types of query datasets.

4.12 Modality contribution to the integrated clustering

We assess the contribution of different modalities to clustering by measuring the agreement between single-modality clustering and multi-modalities cell clustering. For each cell, the normalized consistency ratio of the nearest neighbors in the single modal clustering and multi-modalities clustering is used to represent contribution of the modal for the final integrated clustering.

4.13 Regulatory network inference from scRNA-seq datasets

GRNBoost2 [46] is used to infer the regulatory network from scRNA-seq datasets and is one of the regression-based methods for regulatory network inference. Weighted regulatory links between genes and transcription factors are provided from GRNBoost2. The weights of shared links from different data are compared to indicate the regulatory network retention.

4.14 Correlation of expression fold changes between raw and batch-corrected data

For each cell type, expression fold changes of genes and proteins are calculated against all other cells using FoldChange function in Seurat. Pearson correlation coefficient is used to measure linear correlations of fold changes between raw and batch-corrected data.

4.15 Cell type annotation

To annotate cells in trimodal PBMC atlas, we utilized FindTransferAnchors and TransferData function in Seurat to perform label transfer. We take the PBMC data from Satiji lab as reference and use 'cca' reduction method. For cells

without raw RNA expression, we first utilized ATAC data to create a gene activity matrix using GeneActivity function in Signac [47]. The gene activity matrix is subsequently used for label transfer.

4.16 Evaluation metrics

To evaluate the performance of MIDAS and SOTA tools on multi-omics integration, we utilize metrics from scIB on batch correction and biological conservation, and also propose our own metrics on modality alignment to better evaluate mosaic integration, extending scIB to scMIB. Since MIDAS generates dimensionality reduction result and batch corrected modalities as well, the benchmarking is performed on both latent space and observation space. Note that for batch corrected modalities on the observation space, Seurat is used for data preprocessing and PCA is used for dimensionality reduction.

To get the final overall score, we first get a batch score, a biology score and a modality score, to represent the batch correction, biological conservation and modality alignment performances, respectively. The batch score is the average of Graph iLISI, Graph connectivity and kBet scores; the biology score is the average of NMI, ARI, Isolated label F1 and Graph cLISI; the modality score is the average of Modality ASW, FOSCTTM, Label transfer F1, AUROC, RNA Pearson's *r* and ADT Pearson's *r*. For scIB, the overall score is the weighted summation of the batch score with a weight of 0.4 and the biology score with a weight of 0.6. For scMIB, the overall score is the weighted summation of the batch score with a weight of 0.3, the modality score with a weight of 0.3 and the biology score with a weight of 0.4. Note for scMIB, all the batch metrics and biology metrics are considered both in the embedding space and the feature space.

4.16.1 Batch correction metrics

Graph iLISI. The graph iLISI (local inverse Simpson's index [48]) metric is extend from the iLISI, which is used to measure the batch mixing degree. The iLISI scores are computed based on kNN graphs by computing the inverse Simpson's index for diversity. The scores estimate the effective number of batches present in the neighborhood. iLISI ranges from 1 to N, where N equals the number of batches. Scores close to the real batch numbers denotes good mixing. However, typical iLISI score is not applicable to graph-based outputs. scIB proposed the Graph iLISI, which utilizes a graph-based distance metric to determine the nearest neighbor list and avoids skews on graph-based integration outputs. The graph iLISI scores are scaled from 0 to 1, indicating perfect separation and perfect mixing, respectively.

Graph connectivity. The Graph connectivity is proposed by scIB to inspect whether cells with same label are connected in the kNN graph of cells. For each label c, we get the largest connected graph component (LCC) of c-labeled cells and divide the LCC size by the c-labeled cells population size to represent the graph connectivity for cell label c. Then, we calculate the connectivity values for all labels and take the average as the total Graph connectivity. The score ranges from 0 to 1. The score 1 means that all cells with the same cell identity from different batches are connected in the integrated kNN graph, which also indicates the perfect batch mixing and vice versa.

kBET. The kBET (k-nearest neighbor batch-effect test [49]) is used to measure the batch mixing at the local level of the k-nearest neighbors. Certain fractions of random cell are repeatedly selected to test whether local label distributions are statistically similar to global label distributions (null hypothesis). The kBET value is the rejection rate over all tested neighborhoods and values close to zero represent that the batches are well mixed. scIB adjusts the kBET with a diffusion-based correction to enable compare unbiased comparison on graph-based and non-graph-based integration results.

4.16.2 Biological conservation metrics

NMI. The NMI (Normalized Mutual Information) measures the similarity between two clustering results, namely predefined cell type labels and clustering result based on dimensionality reduction representations in our scenario. Optimized Louvain clustering is used here according to scIB. The NMI scores are scaled to [0,1] and values 0 and 1 correspond to no mutual information and perfect correlation, respectively.

ARI. The ARI (Adjusted rand index) also measures the overlap of two clustering. The Rand Index [50] consider not only cell pairs that are assigned in the same clusters but also ones in different clusters in the predicted (Louvain clustering) and true clusters (cell types). Raw RI values are corrected for chance into the ARI scores. ARI of 1 represent perfect match and ARI of 0 represent random labeling.

Isolated label F1. scIB proposes the Isolated label F1 score to evaluate the performance of integration tools specifically focusing on cells that are uncommon or infrequent in all batches. Cell labels that only presented in the least number of

batches are identified as isolated labels. The F1 score for measuring the clustering performance on isolated labels are defined as isolated label F1 scores. The value 1 represent that the integration handles perfectly on the uncommon cell types across different batches, vice versa.

Graph cLISI. The Graph cLISI is similar to Graph iLISI but focus on cell labels rather than batch labels. Unlike iLISI, cLISI values the separation of groups while iLISI highlights the mixing of groups. The graph-adjusted cLISI is scaled to [0,1] with value 0 corresponding to low cell-type separation and value 1 corresponding to strong cell-type separation.

4.16.3 Modality alignment metrics

Modality ASW. The ASW (average silhouette width [51]) compare the distance between in-cluster cell distances and between-cluster cell distances. Let L_s represents the average distance of a cell to its same-cluster siblings and L_n represents the average distance of the cell to its near-cluster relatives, then $(L_n - L_s)/\max(L_n, L_s)$ represent the silhouette width of the cell. The ASW denotes the averaged SW scores of all cells. In the modality alignment scenario, we first get three latent representations by inputting the three modalities individually, then calculate the Modality ASW scores by comparing the labels of one modality to the demensionality reduction result derived from another modality. Each pair of the three modalities are compared and the average value is taken as the final Modality ASW.

FOSCTTM. The FOSCTTM (fraction of samples closer than the true match [52]) measures the consistency of two different embedding systems for the same cells. The FOSCTTM score is calculated as

$$\begin{aligned} FOSCTTM &= \frac{1}{2n}\left(\sum_i \frac{a^i}{n} + \sum_i \frac{b^i}{n}\right) \\ a^i &= |\{j \mid d(x_i, y_j) < d(x_i, y_i)\}| \\ b^i &= |\{j \mid d(x_j, y_i) < d(x_i, y_i)\}| \end{aligned} \tag{32}$$

where n is the number of cells, i and j are the cell indices, x and y are the cell embeddings from the two different embedding systems, d is the Euclidean distance, a_i and b_i are the number of cells in one embedding system that are closer to the cell i than itself from the other embedding system. Under the modality alignment scenario, we first get three latent spaces by inputting each modality individually to the trained model, then calculate the FOSCTTM for each modality pair, and finally take the averaged value as the final FOSCTTM score. Higher FOSCTTM scores indicate that higher consistency among the three modalities' latent spaces.

Label transfer F1. To inspect whether the latent space of each modality conserved the biological states from the aligned space, we test whether labels can be transferred from one modality to another without any bias. For each pair of modalities, we first build a kNN graph for the two dimensionality reduction results and then transfer labels from one modality to the other based on the nearest neighbors. The transferred labels are compared to the original labels by F1 scores, which is defined as the Label transfer F1. Similarly, we take the average F1 score from all comparison pairs as the final Label transfer F1 score.

ATAC AUROC, RNA Pearson's *r*, and ADT Pearson's *r*. For mosaic integration tasks, MIDAS can generate count matrices for missing modalities by latent modality alignment and data imputation, which give us an opportunity to measure the modality alignment performance on the observed space. We use one modality alone, or two modalities together, as the input, to impute the rest modalities. There are in total 12 different imputation patterns, namely from ATAC to ADT, ATAC to RNA, ADT to ATAC, ADT to RNA, RNA to ATAC, RNA to ADT, ATAC+ADT to RNA, ADT+RNA to ATAC and ATAC+RNA to ADT. The imputed RNA and ADT count matrices are compared to original matrices by Pearson coefficient. The ATAC peaks are evaluated by AUROC scores since the imputed peak values with the range [0,1] are required to be transferred into binary {0, 1} values in order to compare to the original peak matrices. Average scores of different imputation patterns are taken as the final alignment scores. The higher the ATAC AUROC, RNA Pearson's *r* and the ADT Pearson's *r*, the more accurate the model imputes.

4.17 Data availability

All datasets of human PBMCs were publicly available (Supplementary Table 5). Count matrices of gene UMIs, ATAC fragments and antibody-derived tags were downloaded for data analysis.

DOGMA-seq dataset. DOGMA-seq dataset contains four batches profiled by DOGMA-seq, which measures RNA, ATAC and ADT data simultaneously. Trimodal data of this dataset were obtained from Gene Expression Omnibus (GEO) [53] under accession ID GSE166188 [2].

TEA-seq dataset. TEA-seq dataset contains five batches profiled by TEA-seq, which measure RNA, ATAC and ADT data simultaneously. Trimodal data of these batches were obtained from GEO under accession ID GSE158013 [3].

TEA Multiome dataset. TEA Multiome dataset measuring paired RNA and ATAC data was obtained from GEO under accession ID GSE158013 [3]. This dataset contains two batches profiled by 10x Chromium Single Cell Multiome ATAC + Gene Expression.

10X Multiome dataset. The 10X Multiome dataset measuring paired RNA and ATAC data was collected from 10x Genomics (https://www.10xgenomics.com/resources/datasets/) [54–57].

ASAP dataset. The ASAP dataset was obtained from GEO under accession ID GSE156473 [2]. Two batches profiled by ASAP-seq include ATAC and ADT data, and the other two batches profiled by CITE-seq measure RNA and ADT data.

WNN dataset. The WNN dataset measuring paired RNA and ADT data was obtained from https://atlas.fredhutch.org/nygc/multimodal-pbmc/ [13]. This dataset was profiled by CITE-seq. We selected the eight PBMC batches before administration of an HIV vaccine for integration.

4.18 Data preprocessing

The count matrices of RNA and ADT were processed via the Seurat package (v4.1.0) [13]. The ATAC fragment files were processed using the Signac package (v1.6.0) [47] and peaks were called via MACS2 [58]. We performed quality control separately for each batch. Briefly, metrics of detected gene number per cell, total UMI number, percentage of mtRNA reads, total protein tag number, total fragment number, TSS score and nucleosome signal were evaluated. We manually checked the distributions of these metrics and set customized criteria to filter low-quality cells in each batch. The number of cells that passed quality control in each batch is shown in Supplementary Table 5.

We adopted common normalization strategies across all batches for RNA, ADT and ATAC respectively. Specifically, for RNA data, UMI count matrices are normalized and log-transformed using the NormalizeData function in Seurat; for ADT data, tag count matrices are centered log-ratio (CLR)-normalized using the NormalizeData function in Seurat; for ATAC data, fragment matrices are term frequency inverse document frequency (TF-IDF) normalized using the RunTFIDF function in Signac.

To integrate batches profiled by various technologies, we need to create a union of features for RNA, ADT and ATAC data, respectively. For RNA data, firstly, low-frequency genes are removed based on gene occurrence frequency across all batches; then we selected 4000 high variable genes (HVGs) using the FindVariableFeatures function with default parameters in each batch; the union of these HVGs are ranked using the SelectIntegrationFeatures function and the top 4000 genes are selected. In addition, we also retained genes that encode the proteins targeted by the antibodies. For ADT data, the union of antibodies in all batches are retained for data integration. For ATAC data, we used reduce function in Signac to merge all intersecting peaks across batches and then re-calculated the fragment counts in the merged peaks. The merged peaks are used for data integration.

The unions of RNA, ADT and ATAC features are used for MIDAS input. The input data for MIDAS are UMI counts for RNA data, tag counts for ADT data and binarized fragment count for ATAC data. Counts of features that are not included in batches are set to 0.

4.19 Implementation of MIDAS

We implement the architecture of MIDAS with PyTorch [59]. We set the sizes of the shared hidden layers of different modality encoders to 1024-128, set the sizes of the shared hidden layers of different modality decoders to 128-1024, and set the sizes of the biological state and technical noise hidden variables to 32 and 2, respectively. Each hidden layer is implemented by four functions: Linear, LayerNorm, Mish, and Dropout. For all tasks, we split the training data into a proportion of 95/5 for training/validation. To train the model, we set the minibatch size to 256, and use the AdamW [60] optimizer with a learning rate of 10^{-4} to implement SGD. Early stopping is used to terminate training.

4.20 Implementation of comparing methods

As there are few methods for trimodal mosaic integration, we compare MIDAS to recent methods in trimodal rectangular integration. Rectangular integration is a simpler case of mosaic integration, where all modalities are complete. Since there are a few methods directly applicable to rectangular integration tasks involving ATAC, RNA, and ADT, we also divide rectangular integration into two steps, *i.e.*, batch correction for each modality independently, and modality fusion for all batch-corrected modalities. Then, we additionally combine different batch correction and modality fusion methods to achieve rectangular integration, resulting in 9 different strategies in total. In each modality, since these strategies cannot handle missing features, we only use the feature intersection of different batches of the preprocessed data. In addition, here all the generated low-dimensional representations have the same size as the biological states inferred by MIDAS, which are 32 dimensions.

BBKNN+average. BBKNN [61] is used for batch correction (embedding space) and graph averaging is used for modality fusion. For each batch, we employ functions from the Seurat package to perform dimensionality reduction on the count data. We first use RunTFIDF and RunSVD functions to obtain the low-dimensional representation of ATAC, and then use NormalizeData, ScaleData, and RunPCA functions to obtain the low-dimensional representations of RNA and ADT, respectively. For the obtained low-dimensional representation of each modality, we use the bbknn function of the Scanpy package to remove the batch effect and obtain an similarity graph. Finally, we average the similarity graphs of different modalities to obtain the output.

Harmony+WNN. Harmony [48] is used for batch correction (embedding space) and WNN [13] is used for modality fusion. We use the same processing method as BBKNN+average to obtain low-dimensional representations of different batches of ATAC, RNA, and ADT respectively. For the obtained low-dimensional representation of each modality, we employ the RunHarmony function of the Harmony package to remove batch effects. Then, we use Seurat's FindMultiModalNeighbors function, *i.e.*, the WNN algorithm, to fuse the low-dimensional representations of different modalities to obtain the graph output.

LIGER+WNN. LIGER [62] is used for batch correction (embedding space) and WNN is used for modality fusion. For each batch, we employ Seurat's RunTFIDF and ScaleData functions for ATAC normalization and its NormalizeData and ScaleData functions for RNA and ADT normalization. For each modality, we then use RunOptimizeALS and RunQuantileNorm functions of the LIGER package for dimensionality reduction and batch effect removal. Finally, we use the WNN algorithm FindMultiModalNeighbors function to fuse the low-dimensional representations of different modalities to obtain the graph output.

MOFA+. MOFA+ [20] is used for simultaneous batch correction (embedding space) and modality fusion. We first use the same processing method as LIGER+WNN to normalize each modality separately, and then use run_mofa and get_factors functions of the MOFA+ package to achieve simultaneous batch effect removal and modality fusion on the normalized data, obtaining low-dimensional representations output.

PCA+WNN. PCA is used for batch correction (embedding space) and WNN is used for modality fusion. We use the same processing method as BBKNN+average to obtain low-dimensional representations of different batches of ATAC, RNA, and ADT respectively. Then, we use the WNN algorithm FindMultiModalNeighbors function to fuse the low-dimensional representations of different modalities to obtain the graph output.

Scanorama-embed+WNN. Scanorama [62] is used for batch correction (embedding space) and WNN is used for modality fusion. For each modality, we employ integrate function from the Scanorama package for dimensionality reduction and batch effect removal. Then, we use the WNN algorithm FindMultiModalNeighbors function to fuse the low-dimensional representations of different modalities to obtain the graph output.

Scanorama-feat+WNN. Scanorama is used for batch correction (feature space) and WNN is used for modality fusion. For each modality, we perform batch correction using correct function of the Scanorama package. For the batch-corrected count data, we then use the PCA+WNN strategy to get the graph output.

Seurat-CCA+WNN. Seurat's CCA [12] is used for batch correction (feature space) and WNN is used for modality fusion. For each modality, we use Seurat's FindIntegrationAnchors function (reduction = "cca"), *i.e.*, the CCA algorithm, to anchor different batches, and use its IntegrateData function to correct batch effects. For the batch-corrected count data, we then use the PCA+WNN strategy to get the graph output.

Seurat-RPCA+WNN. Seurat's RPCA [12] is used for batch correction (feature space) and WNN is used for modality fusion. It uses the same strategy as Seurat-CCA+WNN, except that FindIntegrationAnchors function is applied with reduction = "rpca".

References

[1] Jeongwoo Lee, Do Young Hyeon, and Daehee Hwang. Single-cell multiomics: Technologies and data analysis methods. *Experimental & Molecular Medicine*, pages 1–15, September 2020. ISSN 2092-6413. doi: 10.1038/s12276-020-0420-2.

[2] Eleni P. Mimitou, Caleb A. Lareau, Kelvin Y. Chen, Andre L. Zorzetto-Fernandes, Yuhan Hao, Yusuke Takeshima, Wendy Luo, Tse-Shun Huang, Bertrand Z. Yeung, Efthymia Papalexi, Pratiksha I. Thakore, Tatsuya Kibayashi, James Badger Wing, Mayu Hata, Rahul Satija, Kristopher L. Nazor, Shimon Sakaguchi, Leif S. Ludwig, Vijay G. Sankaran, Aviv Regev, and Peter Smibert. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nature Biotechnology*, pages 1–13, June 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-00927-2.

[3] Elliott Swanson, Cara Lord, Julian Reading, Alexander T Heubeck, Palak C Genge, Zachary Thomson, Morgan DA Weiss, Xiao-jun Li, Adam K Savage, Richard R Green, Troy R Torgerson, Thomas F Bumol, Lucas T Graybuck, and Peter J Skene. Simultaneous trimodal single-cell measurement of transcripts, epitopes, and chromatin accessibility using TEA-seq. *eLife*, 10:e63632, April 2021. ISSN 2050-084X. doi: 10.7554/eLife.63632.

[4] Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K. Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Simultaneous epitope and transcriptome measurement in single cells. *Nature Methods*, 14(9):865–868, September 2017. ISSN 1548-7105. doi: 10.1038/nmeth.4380.

[5] Song Chen, Blue B. Lake, and Kun Zhang. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nature Biotechnology*, 37(12):1452–1457, December 2019. ISSN 1546-1696. doi: 10.1038/s41587-019-0290-0.

[6] Guoqiang Li, Yaping Liu, Yanxiao Zhang, Naoki Kubo, Miao Yu, Rongxin Fang, Manolis Kellis, and Bing Ren. Joint profiling of DNA methylation and chromatin architecture in single cells. *Nature Methods*, 16(10):991–993, October 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0502-z.

[7] Sai Ma, Bing Zhang, Lindsay M. LaFave, Andrew S. Earl, Zachary Chiang, Yan Hu, Jiarui Ding, Alison Brack, Vinay K. Kartha, Tristan Tay, Travis Law, Caleb Lareau, Ya-Chieh Hsu, Aviv Regev, and Jason D. Buenrostro. Chromatin Potential Identified by Shared Single-Cell Profiling of RNA and Chromatin. *Cell*, 183(4):1103–1116.e20, November 2020. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2020.09.056.

[8] Chenxu Zhu, Yanxiao Zhang, Yang Eric Li, Jacinta Lucero, M. Margarita Behrens, and Bing Ren. Joint profiling of histone modifications and transcriptome in single cells from mouse brain. *Nature Methods*, pages 1–10, February 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01060-3.

[9] Evgenij Fiskin, Caleb A. Lareau, Leif S. Ludwig, Gökcen Eraslan, Feimei Liu, Aaron M. Ring, Ramnik J. Xavier, and Aviv Regev. Single-cell profiling of proteins and chromatin accessibility using PHAGE-ATAC. *Nature Biotechnology*, pages 1–8, October 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-01065-5.

[10] Bingjie Zhang, Avi Srivastava, Eleni Mimitou, Tim Stuart, Ivan Raimondi, Yuhan Hao, Peter Smibert, and Rahul Satija. Characterizing cellular heterogeneity in chromatin state with scCUT&Tag-pro. *Nature Biotechnology*, pages 1–11, March 2022. ISSN 1546-1696. doi: 10.1038/s41587-022-01250-0.

[11] Tim Stuart and Rahul Satija. Integrative single-cell analysis. *Nature Reviews Genetics*, 20(5):257–272, May 2019. ISSN 1471-0064. doi: 10.1038/s41576-019-0093-7.

[12] Tim Stuart, Andrew Butler, Paul Hoffman, Christoph Hafemeister, Efthymia Papalexi, William M. Mauck, Yuhan Hao, Marlon Stoeckius, Peter Smibert, and Rahul Satija. Comprehensive Integration of Single-Cell Data. *Cell*, 177(7):1888–1902.e21, June 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2019.05.031.

[13] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M. Mauck, Shiwei Zheng, Andrew Butler, Maddie J. Lee, Aaron J. Wilk, Charlotte Darby, Michael Zager, Paul Hoffman, Marlon Stoeckius, Efthymia Papalexi, Eleni P. Mimitou, Jaison Jain, Avi Srivastava, Tim Stuart, Lamar M. Fleming, Bertrand Yeung, Angela J. Rogers, Juliana M. McElrath, Catherine A. Blish, Raphael Gottardo, Peter Smibert, and Rahul Satija. Integrated analysis of multimodal single-cell data. *Cell*, May 2021. ISSN 0092-8674. doi: 10.1016/j.cell.2021.04.048.

[14] Mohammad Lotfollahi, Mohsen Naghipourfar, Malte D. Luecken, Matin Khajavi, Maren Büttner, Marco Wagenstetter, Žiga Avsec, Adam Gayoso, Nir Yosef, Marta Interlandi, Sergei Rybakov, Alexander V. Misharin, and Fabian J. Theis. Mapping single-cell data to reference atlases by transfer learning. *Nature Biotechnology*, pages 1–10, August 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-01001-7.

[15] Ricard Argelaguet, Britta Velten, Damien Arnol, Sascha Dietrich, Thorsten Zenz, John C Marioni, Florian Buettner, Wolfgang Huber, and Oliver Stegle. Multi-Omics Factor Analysis—a framework for unsupervised integration of multi-omics data sets. *Molecular Systems Biology*, 14(6):e8124, June 2018. ISSN 1744-4292. doi: 10.15252/msb.20178124.

[16] Adam Gayoso, Zoë Steier, Romain Lopez, Jeffrey Regier, Kristopher L. Nazor, Aaron Streets, and Nir Yosef. Joint probabilistic modeling of single-cell multi-omic data with totalVI. *Nature Methods*, pages 1–11, February 2021. ISSN 1548-7105. doi: 10.1038/s41592-020-01050-x.

[17] Justin Lakkis, Amelia Schroeder, Kenong Su, Michelle Y. Y. Lee, Alexander C. Bashore, Muredach P. Reilly, and Mingyao Li. A multi-use deep learning method for CITE-seq and single-cell RNA-seq data integration with cell surface protein prediction and imputation. *Nature Machine Intelligence*, pages 1–13, October 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00545-w.

[18] Boying Gong, Yun Zhou, and Elizabeth Purdom. Cobolt: Integrative analysis of multimodal single-cell sequencing data. *Genome Biology*, 22(1):351, December 2021. ISSN 1474-760X. doi: 10.1186/s13059-021-02556-z.

[19] Tal Ashuach, Mariano I. Gabitto, Michael I. Jordan, and Nir Yosef. MultiVI: Deep generative model for the integration of multi-modal data, September 2021.

[20] Ricard Argelaguet, Damien Arnol, Danila Bredikhin, Yonatan Deloro, Britta Velten, John C. Marioni, and Oliver Stegle. MOFA+: A statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology*, 21(1):111, May 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-02015-1.

[21] Zhi-Jie Cao and Ge Gao. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. *Nature Biotechnology*, pages 1–9, May 2022. ISSN 1546-1696. doi: 10.1038/s41587-022-01284-4.

[22] Kai Cao, Qiyu Gong, Yiguang Hong, and Lin Wan. uniPort: A unified computational framework for single-cell data integration with optimal transport, February 2022.

[23] Anjun Ma, Adam McDermaid, Jennifer Xu, Yuzhou Chang, and Qin Ma. Integrative Methods and Practical Challenges for Single-Cell Multi-omics. *Trends in Biotechnology*, 38(9):1007–1022, September 2020. ISSN 0167-7799, 1879-3096. doi: 10.1016/j.tibtech.2020.02.013.

[24] Ricard Argelaguet, Anna S. E. Cuomo, Oliver Stegle, and John C. Marioni. Computational principles and challenges in single-cell data integration. *Nature Biotechnology*, pages 1–14, May 2021. ISSN 1546-1696. doi: 10.1038/s41587-021-00895-7.

[25] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, pages 1–7, 2022.

[26] Shujian Yu, Luis G Sánchez Giraldo, and José C Príncipe. Information-theoretic methods in deep neural networks: Recent advances and emerging opportunities. In *IJCAI*, pages 4669–4678, 2021.

[27] Romain Lopez, Adam Gayoso, and Nir Yosef. Enhancing scientific discoveries in molecular biology with deep generative models. *Molecular Systems Biology*, 16(9):e9198, September 2020. ISSN 1744-4292. doi: 10.15252/msb.20199198.

[28] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3116668.

[29] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539.

[30] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & Molecular Medicine*, 50(8):1–14, August 2018. ISSN 2092-6413. doi: 10.1038/s12276-018-0071-8.

[31] Malte D Luecken and Fabian J Theis. Current best practices in single-cell RNA-seq analysis: A tutorial. *Molecular Systems Biology*, 15(6):e8746, June 2019. ISSN 1744-4292. doi: 10.15252/msb.20188746.

[32] Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, November 2019. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000056.

[33] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]*, May 2014.

[34] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [cs, stat]*, September 2020.

[35] Malte D. Luecken, M. Büttner, K. Chaichoompu, A. Danese, M. Interlandi, M. F. Mueller, D. C. Strobl, L. Zappia, M. Dugas, M. Colomé-Tatché, and Fabian J. Theis. Benchmarking atlas-level data integration in single-cell genomics. *Nature Methods*, pages 1–10, December 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01336-8.

[36] Nana H Overgaard, Ji-Won Jung, Raymond J Steptoe, and James W Wells. Cd4+/cd8+ double-positive t cells: more than just a developmental stage? *Journal of leukocyte biology*, 97(1):31–38, 2015.

[37] Dale I Godfrey, Adam P Uldrich, James McCluskey, Jamie Rossjohn, and D Moody. The burgeoning family of unconventional t cells. *Nature immunology*, 16(11):1114–1123, 2015.

[38] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl_1):D258–D261, 2004.

[39] Lam-Ha Ly and Martin Vingron. Effect of imputation on gene network reconstruction from single-cell rna-seq data. *Patterns*, 3(2):100414, 2022.

[40] Françoise Chapuis, Michelle Rosenzwajg, Micael Yagello, Marianne Ekman, Peter Biberfeld, and Jean Claude Gluckman. Differentiation of human dendritic cells from monocytes in vitro. *European journal of immunology*, 27(2):431–441, 1997.

[41] Lei Xiong, Kang Tian, Yuzhe Li, Weixi Ning, Xin Gao, and Qiangfeng Cliff Zhang. Online single-cell data integration through projecting heterogeneous datasets into a common cell-embedding space. *Nature Communications*, 13(1):6118, October 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-33758-z.

[42] Meng Yang, Yueyuxiao Yang, Chenxi Xie, Ming Ni, Jian Liu, Huanming Yang, Feng Mu, and Jian Wang. Contrastive learning enables rapid mapping to multimodal single-cell atlas of multimillion scale. *Nature Machine Intelligence*, pages 1–14, August 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00518-z.

[43] Fan Yang, Wenchuan Wang, Fang Wang, Yuan Fang, Duyu Tang, Junzhou Huang, Hui Lu, and Jianhua Yao. scBERT as a large-scale pretrained deep language model for cell type annotation of single-cell RNA-seq data. *Nature Machine Intelligence*, pages 1–15, September 2022. ISSN 2522-5839. doi: 10.1038/s42256-022-00534-z.

[44] Gido M van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, pages 1–13, 2022.

[45] Mike Wu and Noah Goodman. Multimodal Generative Models for Scalable Weakly-Supervised Learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[46] Thomas Moerman, Sara Aibar Santos, Carmen Bravo González-Blas, Jaak Simm, Yves Moreau, Jan Aerts, and Stein Aerts. Grnboost2 and arboreto: efficient and scalable inference of gene regulatory networks. *Bioinformatics*, 35(12):2159–2161, 2019.

[47] Tim Stuart, Avi Srivastava, Shaista Madad, Caleb A. Lareau, and Rahul Satija. Single-cell chromatin state analysis with Signac. *Nature Methods*, pages 1–9, November 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01282-5.

[48] Ilya Korsunsky, Nghia Millard, Jean Fan, Kamil Slowikowski, Fan Zhang, Kevin Wei, Yuriy Baglaenko, Michael Brenner, Po-ru Loh, and Soumya Raychaudhuri. Fast, sensitive and accurate integration of single-cell data with Harmony. *Nature Methods*, 16(12):1289–1296, December 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0619-0.

[49] Maren Büttner, Zhichao Miao, F. Alexander Wolf, Sarah A. Teichmann, and Fabian J. Theis. A test metric for assessing single-cell RNA-seq batch correction. *Nature Methods*, 16(1):43–49, January 2019. ISSN 1548-7105. doi: 10.1038/s41592-018-0254-1.

[50] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[51] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[52] Ritambhara Singh, Pinar Demetci, Giancarlo Bonora, Vijay Ramani, Choli Lee, He Fang, Zhijun Duan, Xinxian Deng, Jay Shendure, Christine Disteche, and William Stafford Noble. Unsupervised manifold alignment for single-cell multi-omics data. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, BCB '20, pages 1–10, New York, NY, USA, September 2020. Association for Computing Machinery. ISBN 978-1-4503-7964-9. doi: 10.1145/3388440.3412410.

[53] Ron Edgar, Michael Domrachev, and Alex E Lash. Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic acids research*, 30(1):207–210, 2002.

[54] PBMC from a healthy donor (10k), Single Cell Multiome ATAC gene expression Dataset by Cell Ranger ARC 2.0.0. *10X Genomics*, https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-no-cell-sorting-10-k-1-standard-2-0-0, 2021.

[55] PBMC from a healthy donor (3k), Single Cell Multiome ATAC gene expression Dataset by Cell Ranger ARC 2.0.0. *10X Genomics*, https://www.10xgenomics.com/resources/datasets/pbmc-from-a-healthy-donor-no-cell-sorting-3-k-1-standard-2-0-0, 2021.

[56] PBMC from a healthy donor, Single Cell Multiome ATAC gene expression Dataset by Chromium Controller. *10X Genomics*, https://www.10xgenomics.com/resources/datasets/10-k-human-pbm-cs-multiome-v-1-0-chromium-controller-1-standard-2-0-0, 2021.

[57] PBMC from a healthy donor, Single Cell Multiome ATAC gene expression Dataset by Chromium X. *10X Genomics*, https://www.10xgenomics.com/resources/datasets/10-k-human-pbm-cs-multiome-v-1-0-chromium-x-1-standard-2-0-0, 2021.

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

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[60] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv:1711.05101 [cs, math]*, January 2019.

[61] Krzysztof Polański, Matthew D Young, Zhichao Miao, Kerstin B Meyer, Sarah A Teichmann, and Jong-Eun Park. BBKNN: Fast batch alignment of single cell transcriptomes. *Bioinformatics*, 36(3):964–965, February 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz625.

[62] Joshua D. Welch, Velina Kozareva, Ashley Ferreira, Charles Vanderburg, Carly Martin, and Evan Z. Macosko. Single-Cell Multi-omic Integration Compares and Contrasts Features of Brain Cell Identity. *Cell*, 177(7): 1873–1887.e17, June 2019. ISSN 0092-8674. doi: 10.1016/j.cell.2019.05.006.

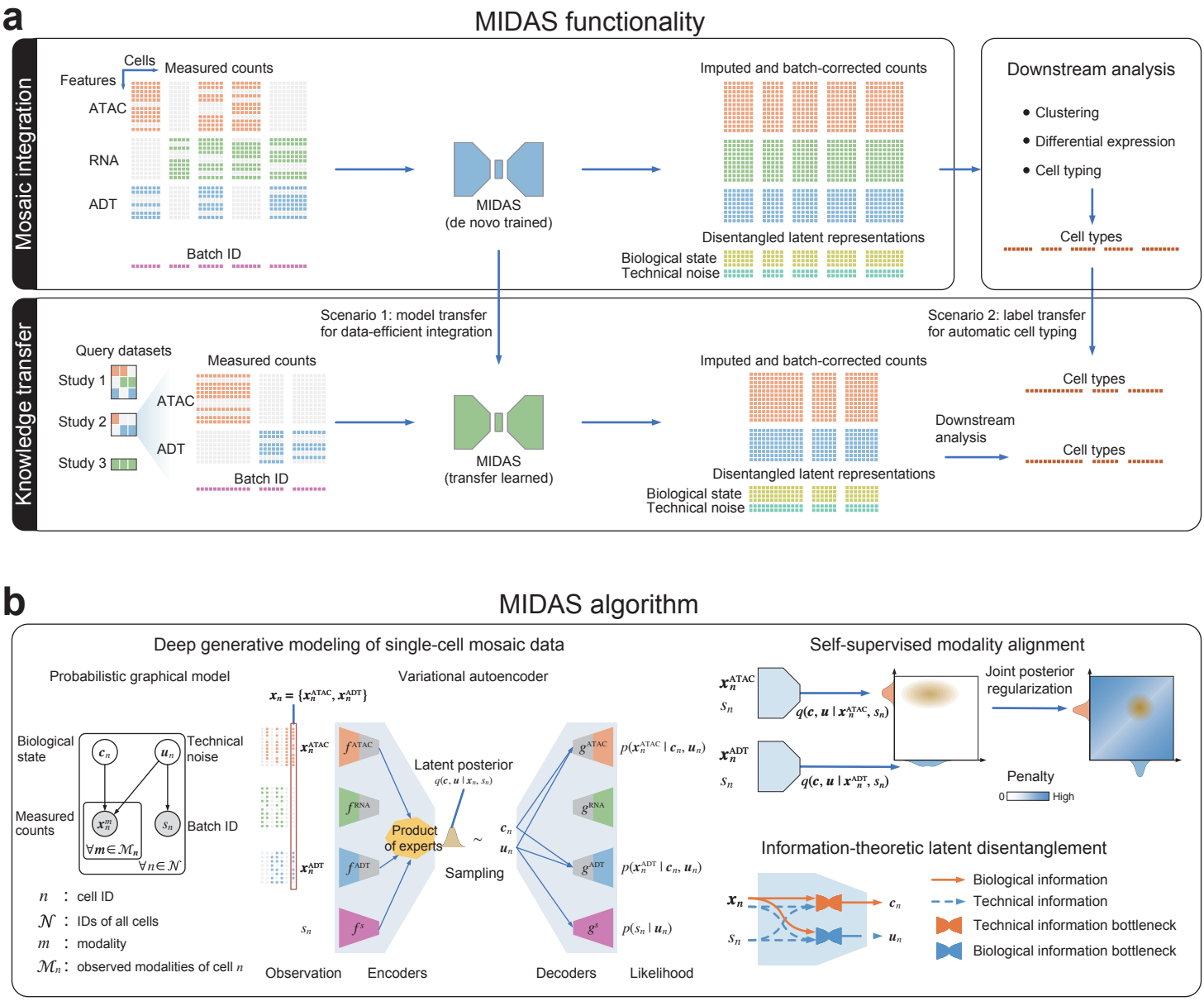

Figure 1. Overview of the MIDAS framework. **a.** Functionality of the MIDAS framework. **b.** MIDAS assumes each cell's measured counts and batch ID are generated from the biological state and technical noise latent variables, and employs the variational autoencoder to implement model learning and latent variable inference. Self-supervised learning is used to align different modalities on latent space through joint posterior regularization, and information-theoretic approaches help to disentangle the latent variables. **c.** Two strategies are developed for MIDAS to achieve reference-to-query knowledge transfer, where the model transfer utilizes pre-trained model for data-efficient integration, and the label transfer reciprocally maps the reference and query datasets onto the latent space for automatic cell typing.

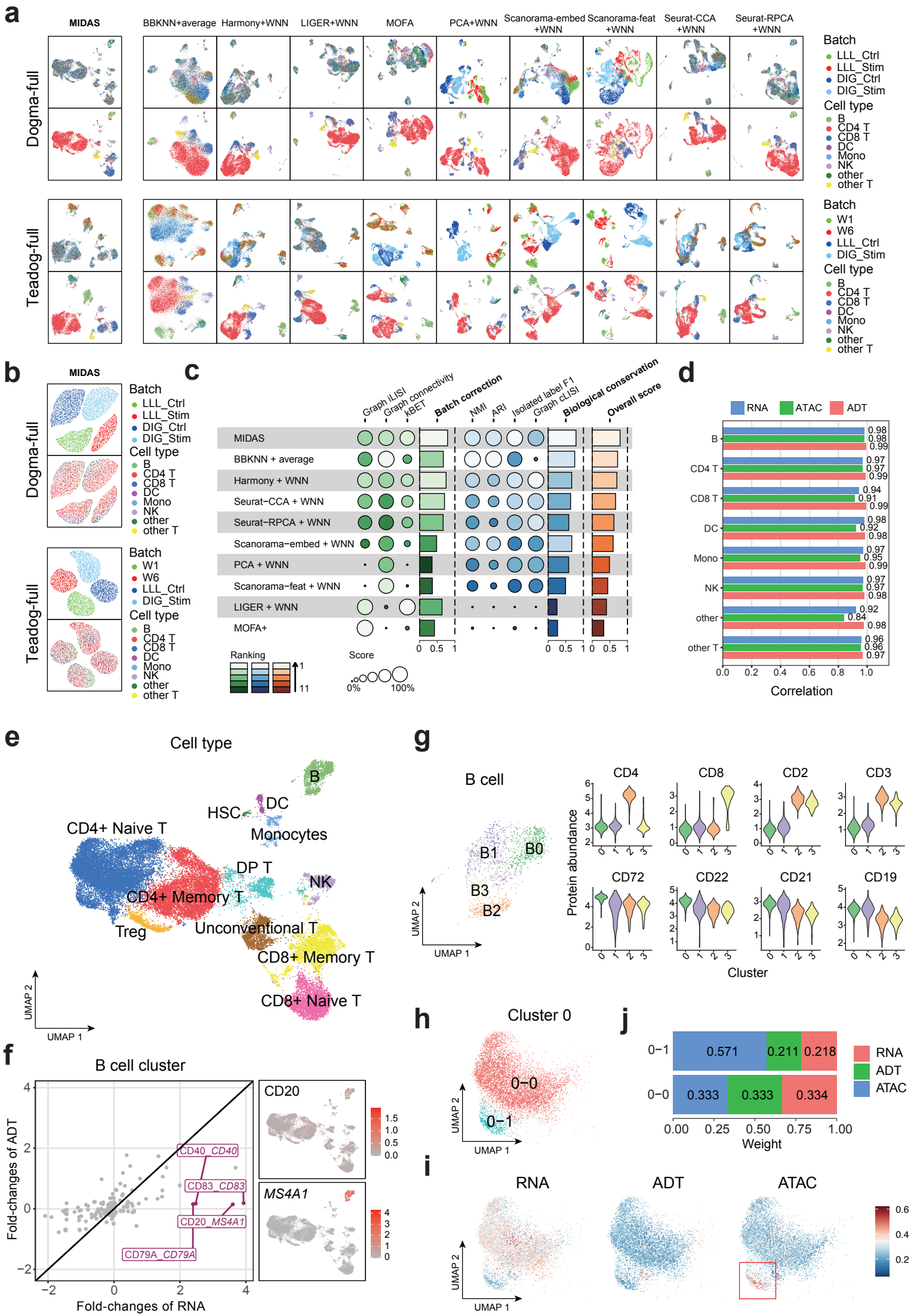
a
MIDAS
BBKNN+average
Harmony+WNN
LIGER+WNN
MOFA
PCA+WNN
Scanorama-embed +WNN
Scanorama-feat +WNN
Seurat-CCA +WNN
Seurat-RPCA +WNN
Dogma-full
Teadog-full
Batch
LLL_Ctrl
LLL_Stim
DIG_Ctrl
DIG_Stim
Cell type
B
CD4 T
CD8 T
DC
Mono
NK
other
other T
Batch
W1
W6
LLL_Ctrl
DIG_Stim
b
c
Graph iLISI
Graph connectivity
kBET
Batch correction
NMI
ARI
Isolated label F1
Graph cLISI
Biological conservation
Overall score
MIDAS
BBKNN + average
Harmony + WNN
Seurat−CCA + WNN
Seurat−RPCA + WNN
Scanorama−embed + WNN
PCA + WNN
Scanorama−feat + WNN
LIGER + WNN
MOFA+
Ranking
Score
0%
100%
d
RNA
ATAC
ADT
Correlation
e
Cell type
B
DC
HSC
Monocytes
CD4+ Naive T
DP T
NK
CD4+ Memory T
Treg
Unconventional T
CD8+ Memory T
CD8+ Naive T
UMAP 2
UMAP 1
f
B cell cluster
Fold-changes of ADT
Fold-changes of RNA
CD40_CD40
CD83_CD83
CD20_MS4A1
CD79A_CD79A
CD20
MS4A1
g
B cell
B1
B0
B3
B2
CD4
CD8
CD2
CD3
CD72
CD22
CD21
CD19
Protein abundance
Cluster
h
Cluster 0
0−0
0−1
j
0−1
0−0
0.571
0.211
0.218
0.333
0.333
0.334
Weight
i
RNA
ADT
ATAC

Figure 2. Evaluation and downstream analysis results obtained with MIDAS on rectangular integration tasks. a. UMAP visualization of cell embeddings obtained by MIDAS and nine other strategies in the dogma-full and teadog-full datasets. The left two panels show inferred latent biological states and the right panels show dimensionality reduction results obtained with the other strategies. **b.** UMAP visualization of latent technical noise inferred by MIDAS in the dogma-full and teadog-full datasets. **c.** scIB benchmarking of performance on the dogma-full rectangular integration task. **d.** Correlation of fold changes in gene/protein abundance and chromatin accessibility between raw and batch-corrected data. **e.** UMAP visualization of the inferred latent biological states of manually annotated cell types. **f.** Expression inconsistencies between proteins and their corresponding genes in B cells. The left panel shows RNA and ADT fold changes and the right panel shows the UMAP visualization of CD20 and *MS4A1* expression. **g.** UMAP visualization of B-cell subclusters (left) and violin plots of protein abundance across subclusters (right). **h.** UMAP plot of $CD4^+$ naive T-cell C0-0 and C0-1 subclusters from the dogma dataset. **i.** Single-cell modality contributions to C0 clustering. The red rectangle highlights the greater contribution of the ATAC modality in cluster C0-1. **j.** Modality contributions to the integrated clustering of C0-0 and C0-1 cells.

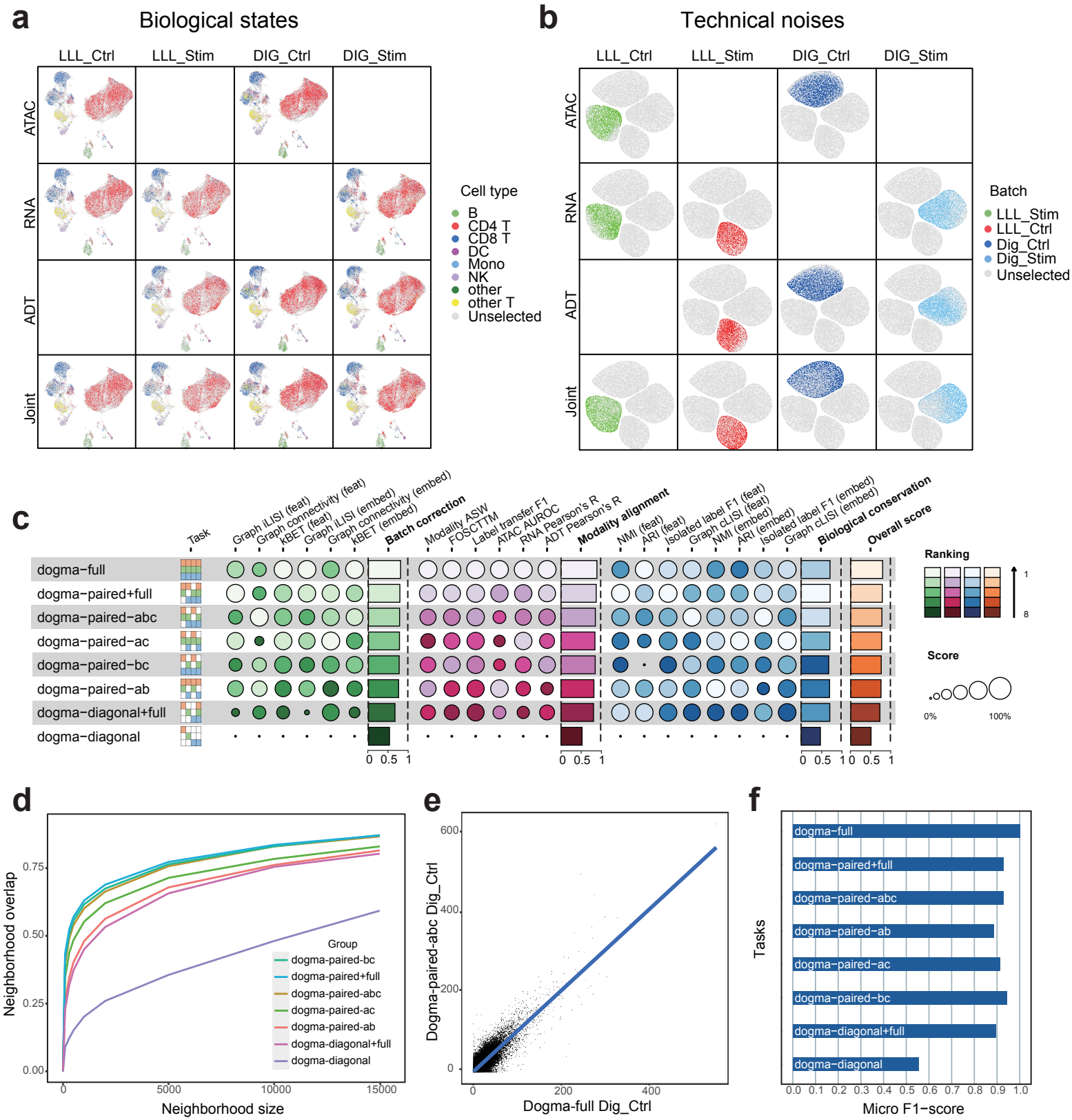

Figure 3. Qualitative and quantitative evaluation of MIDAS performance on mosaic integration tasks. UMAP visualization of the biological states (**a**) and technical noise (**b**) inferred by MIDAS with the dogma-paired-abc dataset. **c.** Benchmarking of MIDAS performance on dogma mosaic integration tasks using our proposed scMIB. **d.** Consistency of dimensional reduction results from different tasks with those from the dogma-full task, measured by the overlap of cells' nearest neighbors. **e.** Consistency of gene regulation links in inferred (dogma-full, DIG_Ctrl batch) and raw (dogma-paired-abc, DIG_Ctrl batch) RNA data. Values represent the regulation importance of gene-transcript factor pairs. **f.** Micro-F1 scores reflecting the consistency of downstream-analyzed cell labels between mosaic tasks and the dogma-full task.

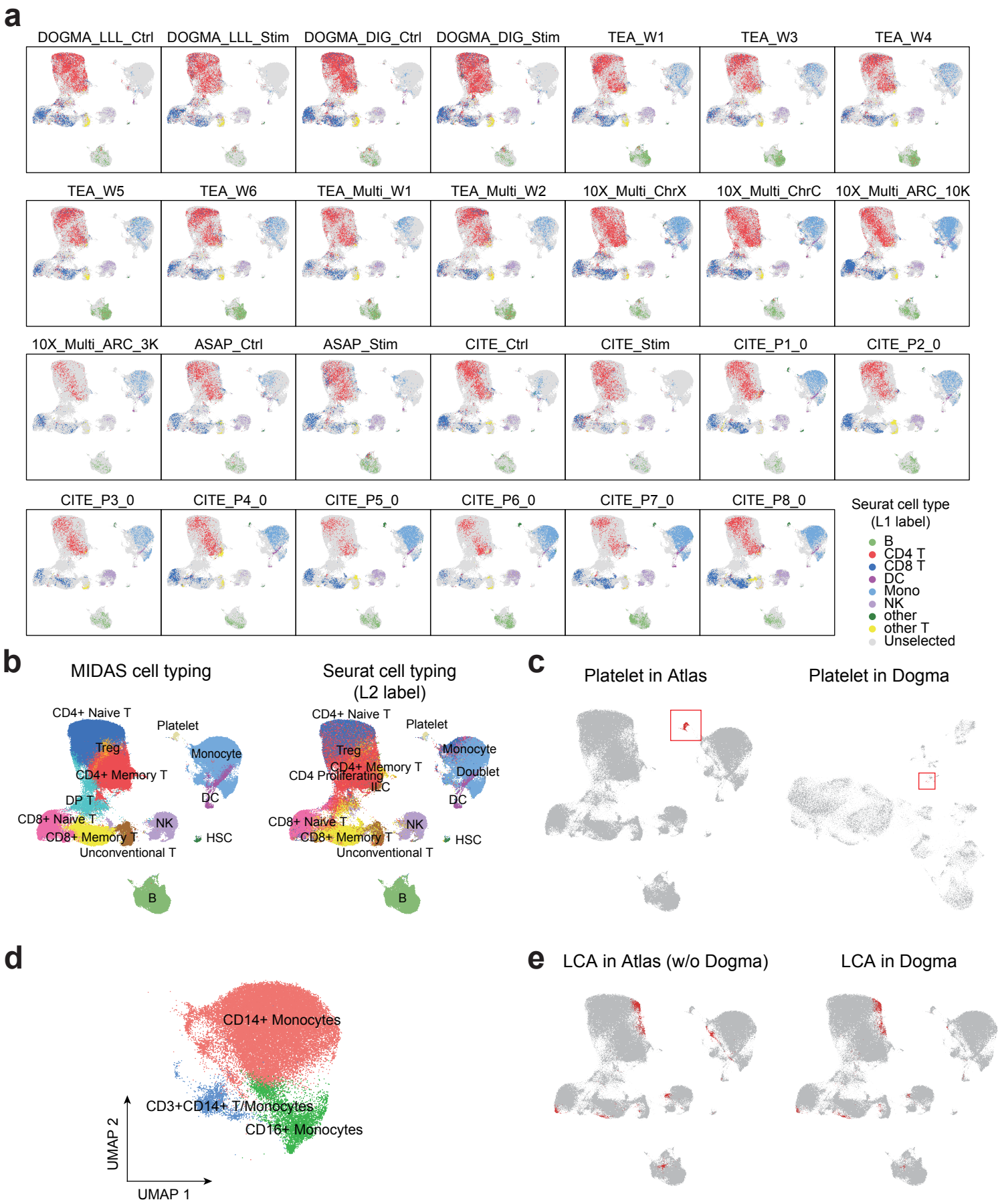

Figure 4. Atlas-level mosaic integration and downstream analysis results obtained with the application of MIDAS to trimodal PBMC data. **a.** UMAP visualization of the biological states inferred by MIDAS with the PBMC mosaic dataset containing 27 batches. Cell type labels are derived from Seurat labeling. **b.** Labels of atlas cells annotated based on clustering and markers (left) and by Seurat (right). **c.** Platelets in the atlas (left) and dogma (right) datasets. **d.** High resolution of monocyte types in the atlas. **e.** Cells with low chromatin accessibility (LCA; red) in the atlas. Left, all LCA cells except those from the dogma dataset; right, LCA cells from the dogma dataset.

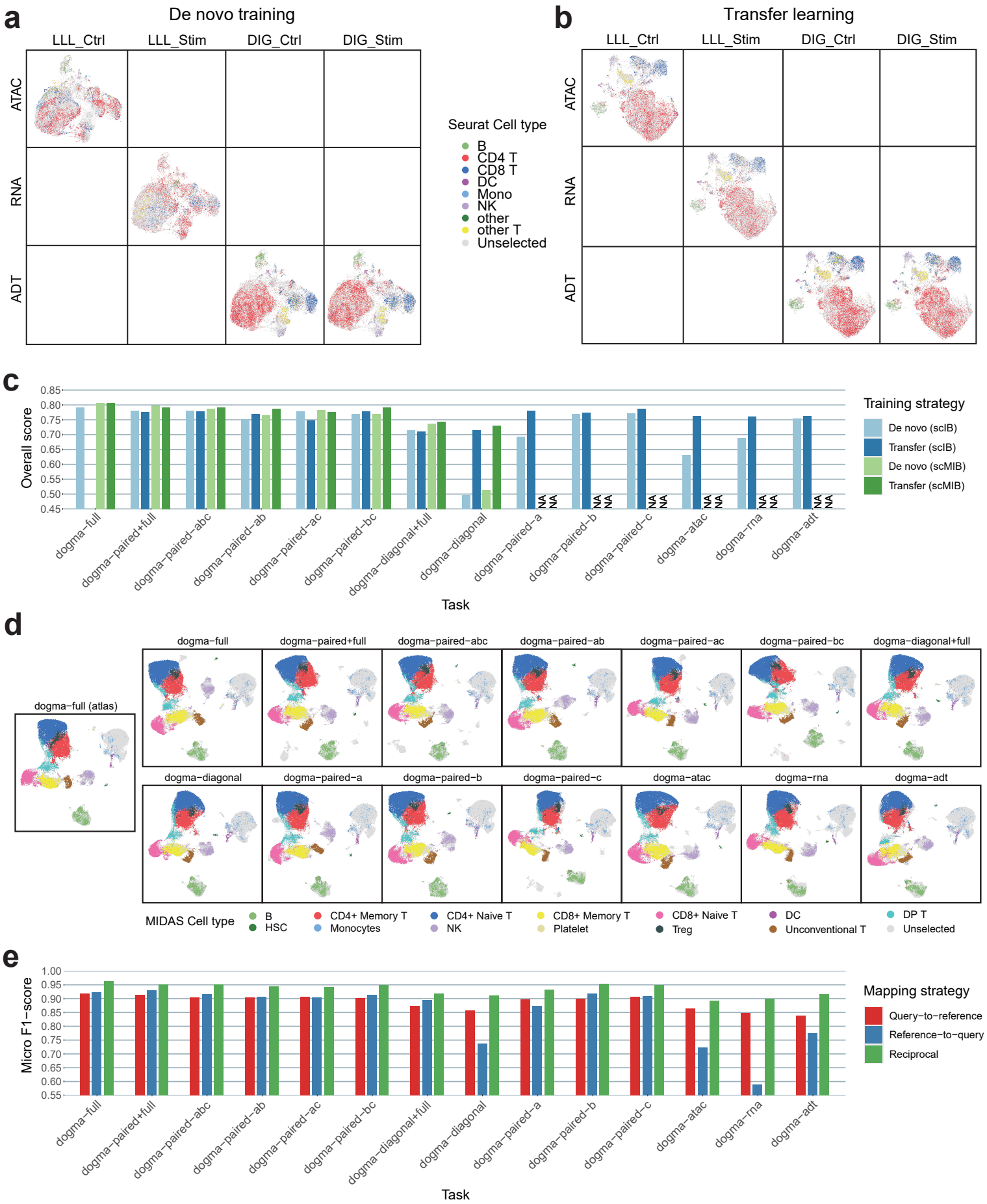

Figure 5. Qualitative and quantitative evaluation of MIDAS on knowledge transfer tasks. UMAP visualization of the biological states inferred by the *de novo* trained (**a**) and transfer-learned (**b**) MIDAS on the dogma-diagonal dataset. **c.** Overall scIB and scMIB scores reflecting transfer-learned and *de novo* trained MIDAS performance on 14 dogma mosaic integration tasks. **d.** UMAP visualization of the biological states obtained by reciprocal reference mapping with 14 dogma mosaic datasets (columns 2–8). Column 1 shows the dogma-full atlas integration. **e.** Label transfer micro-F1 scores representing performance on 14 dogma mosaic integration tasks.