Multimodal weakly supervised learning to identify disease-specific changes in single-cell atlases

Anastasia Litinetskaya[1,4], Maiia Shulman[1,2], Soroor Hediyeh-zadeh[1,2], Amir Ali Moinfar[1,4], Fabiola Curion[1,4], Artur Szałata[1,4], Alireza Omidi[5], Mohammad Lotfollahi[1,3,6†⋆], Fabian J. Theis[1,2,3,4‡⋆]

1 Institute of Computational Biology, Helmholtz Center Munich, Munich, Germany.

2 School of Life Sciences Weihenstephan, Technical University of Munich, Munich, Germany.

3 Wellcome Sanger Institute, Wellcome Genome Campus, Cambridge, UK.

4 School of Computing, Information and Technology, Technical University of Munich, Munich, Germany.

5 Michael Smith Laboratories, University of British Columbia, Canada.

6 Wellcome–MRC Cambridge Stem Cell Institute, University of Cambridge, Cambridge, UK.

⋆ These authors contributed equally to the work.

†Correspondence to:

ml19@sanger.ac.uk

fabian.theis@helmholtz-munich.de

Abstract

Multimodal analysis of single-cell samples from healthy and diseased tissues at various stages provides a comprehensive view that identifies disease-specific cells, their molecular features and aids in patient stratification. Here, we present MultiMIL, a novel weakly-supervised multimodal model designed to construct multimodal single-cell references and prioritize phenotype-specific cells via patient classification. MultiMIL effectively integrates single-cell modalities, even when they only partially overlap, providing robust representations for downstream analyses such as phenotypic prediction and cell prioritization. Using a multiple-instance learning approach, MultiMIL aggregates cell-level measurements into sample-level representations and identifies disease-specific cell states through attention-based scoring. We demonstrate that MultiMIL accurately identifies disease-specific cell states in blood and lung samples, identifying novel disease-associated genes and achieving superior patient classification accuracy compared to existing methods. We anticipate MultiMIL will become an essential tool for querying single-cell multiomic atlases, enhancing our understanding of disease mechanisms and informing targeted treatments.

Introduction

Advances in single-cell technologies have enabled multiomic profiling of thousands of patient samples, providing a holistic view of disease heterogeneity on multiple scales—from individual cells to cell types and patients [1]. These large-scale datasets can facilitate both disease diagnostics and therapeutics [2]. In diagnostics, these multimodal datasets allow for the precise identification of cellular changes that are unique to specific diseases. Researchers can identify biomarkers and cellular behaviors indicative of disease states by analyzing individual cells and their interactions. This level of granularity not only improves the accuracy of diagnostics but also helps in the early detection of diseases, which is crucial for effective treatment. In therapeutics, understanding disease-specific cell states can lead to more targeted and personalized treatment strategies. By identifying the cellular mechanisms and pathways disrupted in disease, researchers can develop therapies targeting these areas, minimizing side effects and improving treatment efficacy.

A significant challenge remains in linking cell-level signals to patient-level phenotypes in an interpretable manner, allowing researchers to understand the underlying cellular processes and mechanisms driving disease phenotypes. Several computational approaches have been developed to predict disease phenotypes at the cellular level [3–7] and at the patient level [8–10]. Concurrently, other approaches prioritize cells exhibiting differential transcriptomic signals [11, 12] or differential compositional signals compared to a reference phenotype (e.g., healthy vs. diseased) [13]. However, these approaches are limited as they model single-cell data based solely on transcriptomics and cannot handle multimodal datasets [3, 6]. Although they provide predictions at the patient level, they fail to effectively link these predictions to the cellular processes driving the disease phenotype [8]. Such approaches also struggle to systematically model technical effects across samples, which is necessary to accurately predict phenotypes and prioritize disease cells free from spurious variations. A recent paper introduced MrVI [14], a model that can deal with batch effects, but it relies heavily on the accuracy of the counterfactual generative modeling with VAEs and does not make use of patient annotations.

To overcome these limitations, we introduce MultiMIL, a multimodal multi-instance learning approach for phenotypic prediction and differential cell prioritization in single-cell multiomics. MultiMIL employs a multiomic data integration strategy using a product-of-expert [15] generative model, providing a comprehensive multimodal representation of cells. These representations are fed into downstream prediction and prioritization modules. The model leverages advances in weakly supervised learning, particularly multiple-instance learning (MIL), to learn patient conditions from single cells by prioritizing phenotype-specific cells through an attention mechanism. The MIL approach allows the model to capture different phenotypic behaviors, from molecular differences to compositional changes upon disease compared to reference phenotypes. MultiMIL can also use latent representations from atlases or foundation models, enhancing its flexibility and utility.

We showcase applications for MultiMIL, enabling efficient multimodal data integration across various datasets, which is necessary to learn robust representations. Using these representations, including pre-trained ones, we demonstrate phenotypic prediction for unseen patients and prioritization of disease-specific cell states by analyzing human peripheral blood mononuclear cells and the Human Lung Cell Atlas. We further demonstrate how the disease states identified with MultiMIL can help discover novel genes associated with the disease.

Learning multimodal cell and patient representations to prioritize phenotype-specific cells

MultiMIL is a deep-learning-based model that allows the integration of multimodal single-cell data and the prediction of sample-level phenotypes from these single-cell measurements. MultiMIL's model consists of two submodules: a variational autoencoder that learns a low-dimensional latent representation of singe-cell data and a classification head that learns to predict sample-level phenotypes from the low-dimensional latent representations (**Fig. 1**a,b). We draw inspiration from the multiple-instance learning (MIL) approach [16, 17], where we model donors as bags and cells as instances belonging to a bag. The classification labels are only known on the bag level but not on the instance level, and we are interested in identifying instances associated with the bag label (**Suppl. Fig. 1**).

The autoencoder module is implemented as encoder-decoder pairs, where each pair corresponds to a modality present in the data (**Fig. 1**a). The encoders output the parameters of the corresponding unimodal marginal distribution, and the joint distribution in the latent space is modeled using the Product of Experts (PoE) [15, 18]. The PoE distribution preserves unique and shared information from the unimodal marginal distributions [18]. The PoE approach also allows MultiMIL to integrate

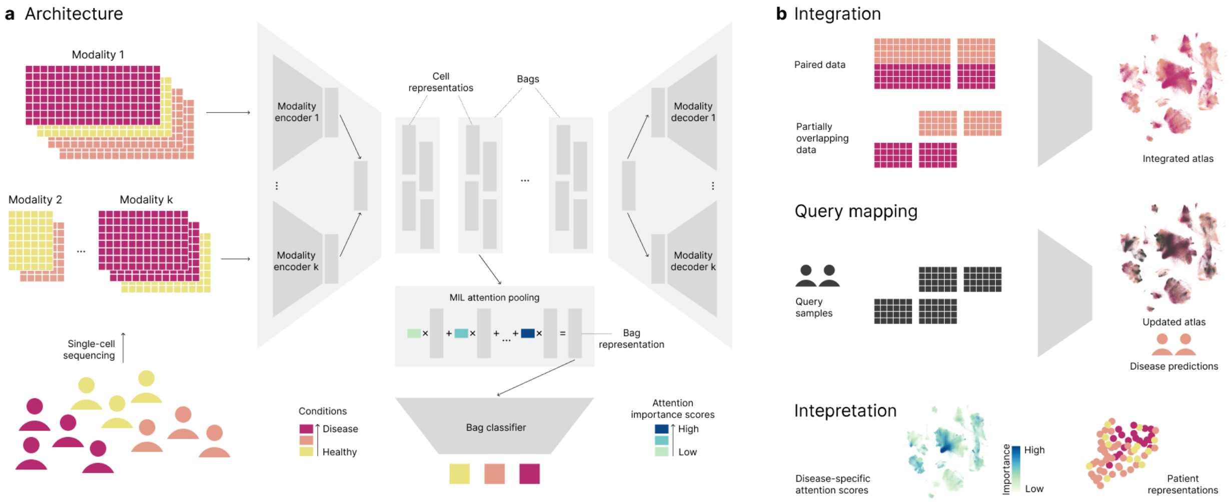


Figure 1: **MultiMIL enables multimodal integration, query mapping and interpretable phenotype prediction.** **(a)** The MultiMIL model accepts paired or partially overlapping single-cell multimodal data across samples with varying phenotypes and consists of pairs of encoders and decoders, where each pair corresponds to a modality. Each encoder outputs a unimodal representation for each cell, and the joint cell representation is calculated from the unimodal representations. The joint latent representations are then fed into the decoders to reconstruct the input data. Cells from the same sample are combined with the MIL attention pooling layer, where cell weights are learned with the attention mechanism, and the sample representations are calculated as a weighted sum of cell representations. The sample representations are then finally fed into the classifier network that learns to predict conditions. **(b)** The key use cases for MultiMIL are the integration of paired and partially overlapping data into reference atlases (top), mapping of query samples onto the reference and prediction of conditions for the new data (middle), and identification of disease-associated cell states with the learned attention weights as well as the construction of disease-informed patient representations (bottom).

paired as well as partially overlapping data (i.e., where the measurements are missing for one or more modalities in part of the data). Additionally, categorical and continuous sample covariates, e.g., batch, can be incorporated into the model to obtain the latent representation disentangled from the specified covariates (see Methods).

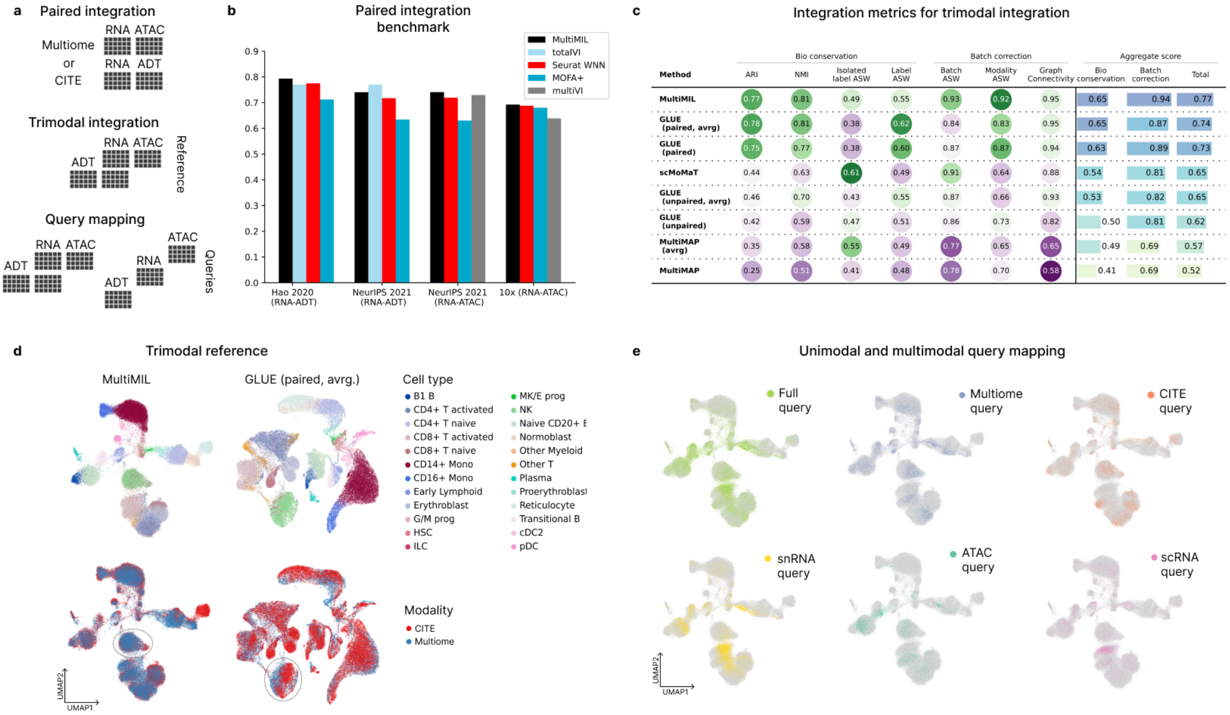
The classification head consists of a MIL aggregator with an attention mechanism and a feed-forward classifier network. The MIL module aggregates the cell-level embeddings into a bag embedding employing attention pooling. During training, the model learns attention weights α_i for each cell i in a bag and then aggregates cell embeddings z_i into a bag representation z_{bag} as weighted sum $\sum_i \alpha_i z_i$. The pooled representation z_{bag} is then fed into a feed-forward network that predicts condition labels.

Ultimately, we are interested in mapping new patients onto the atlases with multiple conditions and predicting the conditions for these patients. To this end, MultiMIL utilizes the scArches transfer-learning approach for query-to-reference mapping [19]. When mapping a new batch of data, we only fine-tune a small portion of the model parameters specific to this batch, allowing for faster and more efficient training compared to *de novo* integration.

MultiMIL provides several ways to interpret the learned attention weights (**Fig. 1**b). Firstly, the higher the weight of a particular cell, the more important the cell was for the prediction. Learning a score for each cell allows us to identify and analyze cell states associated with a particular condition by selecting cells with high attention scores. Additionally, we can obtain sample representations from the model by taking a weighted average of the cells within a sample. These representations of donors in a low-dimensional space are learned from the single-cell measurements and reflect the disease progression better than mean embeddings.

The model is trained on mini-batches, optimizing for the accurate reconstruction, Kullback-Leibler (KL) divergence with monotonic annealing [20, 21] and prediction accuracy. We additionally employ the maximum mean discrepancy loss (MMD) [22, 23] to correct strong batch effects and to make sure that unimodal representations have similar distributions, which is necessary for successful multimodal query-to-reference mapping (see Methods). Due to mini-batching and the deep-learning nature of the model, MultiMIL is fast to train: the integration module takes ca. 10 minutes for a quarter of a million cells and the full model takes ca. 15 minutes for the same number of cells (Table 6).

Users can train the autoencoder module and the classifier head sequentially, separately, or in an end-to-end manner, depending on whether there is a need to integrate the data from scratch or if



	Bio conservation				Batch correction			Aggregate score		
Method	ARI	NMI	Isolated label ASW	Label ASW	Batch ASW	Modality ASW	Graph Connectivity	Bio conservation	Batch correction	Total
MultiMIL	0.77	0.81	0.49	0.55	0.93	0.92	0.95	0.65	0.94	0.77
GLUE (paired, avrg)	0.78	0.81	0.38	0.62	0.84	0.83	0.95	0.65	0.87	0.74
GLUE (paired)	0.75	0.77	0.38	0.60	0.87	0.87	0.94	0.63	0.89	0.73
scMoMaT	0.44	0.63	0.61	0.49	0.91	0.64	0.88	0.54	0.81	0.65
GLUE (unpaired, avrg)	0.46	0.70	0.43	0.55	0.87	0.66	0.93	0.53	0.82	0.65
GLUE (unpaired)	0.42	0.59	0.47	0.51	0.86	0.73	0.82	0.50	0.81	0.62
MultiMAP (avrg)	0.35	0.58	0.55	0.49	0.77	0.65	0.65	0.49	0.69	0.57
MultiMAP	0.25	0.51	0.41	0.48	0.78	0.70	0.58	0.41	0.69	0.52

Figure 2: **MultiMIL integrates paired and partially overlapping modalities and allows unimodal as well as multimodal query-to-reference mapping.** **(a)** Design for the paired integration, trimodal reference building and query mapping. The paired integration benchmark was run on 10x multiome and CITE-seq datasets. The trimodal reference consists of 10x multiome (RNA-ATAC) and CITE-seq (RNA-ADT) data, and the query consists of multiome, CITE-seq and unimodal scATAC-seq and scRNA-seq data. **(b)** A bar plot of overall integration scores for the two CITE-seq (RNA-ADT) and two multiome (RNA-ATAC) datasets comparing MultiMIL, totalVI, Seurat's WNN, MOFA+ and multiVI. **(c)** A table with integration metrics with all the benchmarked methods, showing individual metric scores, averaged bio-conservation and batch-correction scores, and overall scores.**(d)** UMAPs of the reference latent space obtained from the two top-performing models (MultiMIL on the left and paired GLUE, averaged representation on the right), colored by cell type and modality. NK cells appear to be integrated better by MultiMIL, explaining the difference in the overall scores. **(e)** UMAPs of different queries mapped onto the trimodal reference with MultiMIL.

there is already an existing atlas at hand. We will discuss both use cases later. This adaptability makes MultiMIL suitable for a wide range of applications and allows it to integrate seamlessly into existing analytical workflows. We envision MultiMIL as a multi-task tool for multimodal integration, query mapping of new samples, disease prediction for the query donors and identification of disease-associated states.

MultiMIL enables multimodal reference building and query-to-reference mapping

Technologies for paired sequencing enable the joint analysis of two modalities, but this also presents a unique challenge. We need to model both modalities in a way that preserves shared as well as unique information [24, 25]. This work tackles this problem by learning a joint low-dimensional representation for each cell. Due to the modeling of the joint state with the product-of-expert approach, MultiMIL is capable of integrating not only fully paired data but also partially overlapping measurements, for instance, a paired RNA-ATAC dataset and a paired RNA-ADT dataset (**Fig. 2a**). MultiMIL's unique feature is the query mapping of unimodal and multimodal data, which allows the mapping of any combination of modalities onto existing references. In this section, we first compare MultiMIL with the existing methods for paired integration and then demonstrate the trimodal reference building and mapping functionalities.

We benchmarked MultiMIL's performance on paired integration against three state-of-the-art methods on two CITE-seq datasets (NeurIPS 2021 CITE-seq [26], Hao et al. [27]) and two paired RNA-ATAC datasets (NeurIPS 2021 multiome [26], 10x public multiome [28]). Hao et al. dataset comprises PBMCs from eight donors enrolled in an HIV vaccine trial. NeurIPS datasets have bone marrow mononuclear cells from 10 healthy donors, and the second multiome dataset contains PBMCs from one healthy donor and does not have any batch effect. We compared MultiMIL to MOFA+ [29], Seurat v4 WNN [30] on all four datasets, totalVI [31] on CITE-seq datasets and multiVI [32] on the multiome datasets.

To quantitatively evaluate the results, we calculated a subset of the scIB metrics [33] suitable for multimodal integration (see Methods). The metrics address both the conservation of biological signal and batch effect removal. Overall, MultiMIL achieved the highest total score on both paired RNA-ATAC datasets while scoring first and second on the CITE-seq datasets (**Fig. 2a**). TotalVI and Seurat WNN obtained high scores on all datasets, while the score for MultiVI was dataset-dependent (**Suppl. Fig. 2**). MOFA+ failed to remove batch effects present in the original data, resulting in a low batch correction score (**Suppl. Fig. 2**, **Suppl. Fig. 3**).

To demonstrate MultiMIL's ability to perform mosaic integration [24], we integrated Sites 1 and 2 from the NeurIPS 2021 CITE and Neurips 2021 multiome datasets [26]. We compared MultiMIL with GLUE [25], MultiMAP [34] and scMoMaT [35] on this task. We calculated the scIB score on the latent space after performing minimal cell type harmonization between the datasets. We included two Adjusted Silhouette Width (ASW) scores for batch correction: Batch ASW and Modality ASW. This dual-level evaluation of batch and modality mixing allows us to measure the removal of technical biases at a finer scale of individual batches and a coarser scale of modalities simultaneously, aligning with the approach outlined in [36]. For the methods that output one representation per cell per modality, we calculated the metrics once on the original output and once on the averaged representations (denoted "avrg." in **Fig. 2d**).

MultiMIL scored first, and GLUE (paired model, avrg.) scored second on this task. UMAPs of the learned representations are relatively similar for these two methods (**Fig. 2c**). MultiMIL obtained a slightly higher Modality ASW score than GLUE, which is caused, for instance, by better integrated Natural Killer (NK) cells across modalities (**Fig. 2c,d**). scMoMaT scored fourth based on scIB metrics even though the modalities were not well-mixed (**Fig. 2d**, **Suppl. Fig. 4a**). scMoMaT obtained a high Batch ASW score despite not integrating the two modalities. At the same time, we observed that Modality ASW is the lowest for scMoMaT, which aligns with the visual inspection of the UMAPs. Overall, we noted that the models that do take into account the information about which cells are paired (MultiMIL, GLUE paired) performed better than the methods that do not (**Fig. 2d**, **Suppl. Fig. 4**).

When MultiMIL's reference model is trained on multimodal data, our model enables unimodal and multimodal query mapping, where unimodal query modalities can be any of the individual modalities from the multimodal reference. After we build the atlas described above, we map unimodal (i.e., scRNA-seq, snRNA-seq and scATAC-seq) and multimodal (CITE-seq and multiome) queries onto the reference. We calculated scIB metrics using reference and query as two batches to assess the mapping quality. MultiMIL successfully mapped all the queries, obtaining very similar scIB scores for all of them (**Fig. 2e**, **Suppl. Fig. 5c**). Multimodal queries obtained the highest Batch ASW scores, possibly indicating that the batch correction works best for the data modalities present in the reference. We also trained a random forest classifier to transfer the cell types from the reference to the queries and calculated the prediction accuracy. Label transfer worked best for CITE-seq and scRNA data while mapping scATAC-seq seems to be most challenging (**Suppl. Fig. 5c, d**).

Seurat Bridge integration [37] also allows the mapping of scATAC-seq data onto the scRNA-seq reference, so we included it in this experiment. Because the reference in this case is a scRNA-seq-only reference (i.e., not multimodal), we could not directly compare the reference building with the other methods for trimodal reference building. Additionally, Bridge allows visualization of the reference and query on a joint UMAP and label transfer but does not explicitly provide low-dimensional embeddings in the joint reference-query space. Hence, we did not calculate scIB metrics for Seurat Bridge, but we included UMAPs of the reference and the mapped scATAC-seq query in the supplementary figures for visual inspection (**Suppl. Fig. 5a, b**).

To assess the robustness of our model, we performed several experiments benchmarking the model's sensitivity towards the number of shared features, the strength of the integration parameter, the size of the reference and the type of the MMD loss (Methods and **Suppl. Fig. 6**). When the number of shared genes is more than 1,000, MultiMIL can successfully build the reference, but the quality of query mapping increases with the number of shared features. We also observed that the quality of the query mapping slightly increases with bigger references.

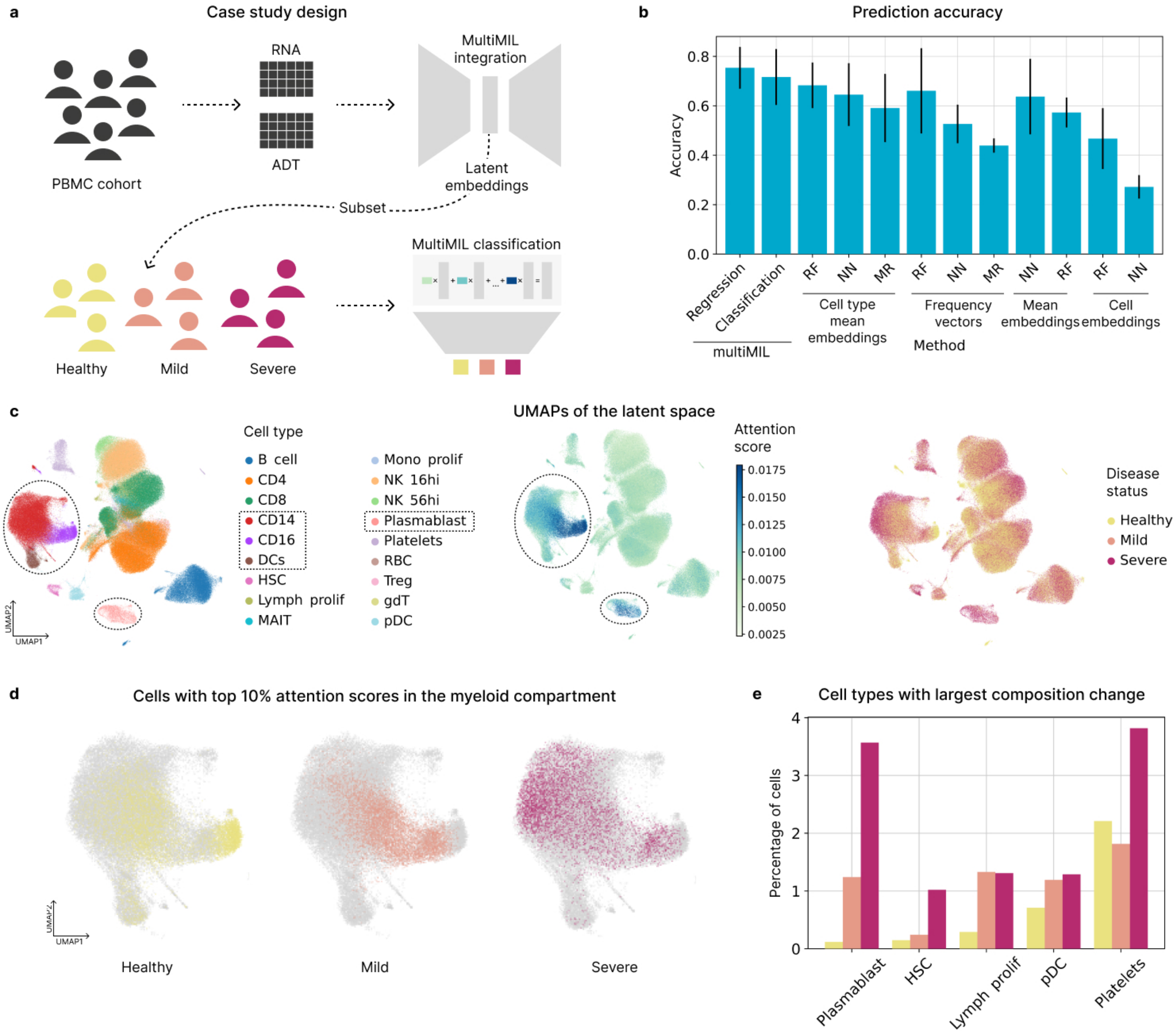


Figure 3: **MultiMIL accurately predicts disease stages from a multimodal PBMC dataset.** **(a)** Case study design. PBMCs were sequenced with CITE-seq (paired RNA and ADT), integrated with MultiMIL's integration module, subset to healthy, mild and severe COVID-19 samples, and used as input to train MultiMIL's classifier network. **(b)** A bar plot showing average accuracies and standard deviations (i.e., the length of an error bar equals two standard deviations) of the five cross-validation runs on the disease-prediction task. MultiMIL was trained in the classification and regression settings. Cell type mean embeddings and frequency vectors were input to the random forest (RF), feed-forward neural network (NN) and multiclass logistic regression (MR) models. Mean embeddings and cell embeddings were input to the RF and NN models. **(c)** UMAPs of the integrated latent space colored by cell type (left), cell attention scores (middle) and condition (right). The myeloid compartment (i.e., CD14, CD16 monocytes and dendritic cells) and plasmablasts have high attention scores. **(d)** UMAPs of the myeloid compartment showing the healthy, mild and severe COVID-19 cells with the top 10% of attention scores for each condition. **(e)** A bar plot showing the top five cell types with the biggest compositional change from healthy to severe COVID-19, including plasmablasts.

MultiMIL accurately predicts disease states and identifies cell states driving the disease progression

In the previous sections, we described how the integration module of MultiMIL performs multimodal integration and query-to-reference mapping. Next, we simultaneously model the multimodal single-cell embeddings and sample-level covariates, such as e.g. disease. To validate our approach of predicting sample-level disease labels from single-cell data, we utilize a CITE-seq peripheral blood mononuclear cell (PBMC) dataset [38]. This large-scale dataset consists of 130 healthy and diseased samples and provides metadata on the progression of COVID-19 stages. First, we integrate scRNA-seq and ADT measurements from all the data points with MultiMIL to obtain a low-dimensional data representation. Then, we subset the data to healthy, mild and severe COVID-19 samples (see Methods) and train the MultiMIL's classifier module to assess the predictive performance on this multiclass classification task and evaluate the interpretability of cell attention scores (**Fig. 3a**).

For MultiMIL, the prediction task can be formulated as either a classification task or a regression task, as we need to model the progression from healthy to mild to severe stages. We compare our model to several baseline models, and MultiMIL outperformed all the baselines in a 5-fold cross-validation experiment (**Fig. 3b**), achieving an accuracy of 75% for the regression model and 72% for the classification model.

The baseline prediction models include a random forest, feed-forward neural net and multiclass regression. Approaches utilizing single-cell data for phenotypic prediction often rely on (pseudo-)bulk data [7, 39], so we included a range of pseudo-bulk baselines in our comparison. Since MIL models generally fall between models that make predictions on the instance (i.e., single-cell) level and models that make predictions on the bag (i.e., bulk) level, we also include cell-level baselines (**Fig. 3b**, Methods). The mean embedding of a sample is the mean of cell embeddings belonging to this sample, and cell type mean embeddings are calculated as the mean of cell embeddings per cell type and concatenated per sample. Frequency vectors are calculated as relative frequencies of cell types present in each sample. For cell embeddings, the input to the models was the cell embeddings from the integrated space, and the prediction was made for each cell. We note that cell type mean embeddings and frequency vectors are supervised since the cell type labels are required, while MultiMIL, mean embeddings and cell embeddings are not.

To ensure that MultiMIL prediction performance is consistent independently of the learned latent embedding, we also trained a totalVI [31] model in the same setting. We observed that the quality of the embeddings is comparable between the two models (**Fig. 3c**, **Suppl. Fig. 8b,c**) and that the

MultiMIL also outperforms other baselines when trained on totalVI embeddings (**Suppl. Fig. 8a**). We also tested MultiMIL on a binary classification task, predicting healthy vs. COVID-19, and in a more challenging multiclass task, predicting healthy and all five stages of COVID-19. In all the experiments, our model outperformed other baselines or performed on par with supervised cell type mean-embedding baselines (**Suppl. Fig. 8a**).

When analyzing diseased samples, we are interested in identifying cell states affected by the disease. By utilizing the cell-attention module, our model learns a weight for each cell, where higher weights directly correspond to cell states associated with the condition. For visualizations and further analysis, we selected the classification formulation of the model since it provided more robust results discussed later (**Suppl. Fig. 9a**). We also only take into account cells with the 10% highest scores per condition, as these cells are most strongly associated with the disease. We observe in **Fig. 3c** that cell types with the highest attention scores are monocytes, dendritic cells (DCs), plasmablasts, and platelets. We first examine the myeloid compartment (**Fig. 3d**) and notice a trajectory of highlighted CD14 monocytes from healthy and mild to severe, indicating a mean shift in expression levels between different stages. Similarly, we find distinct populations of highlighted healthy and mild CD16 monocytes, confirming that the signal learned with MultiMIL aligns with previous studies reporting strong changes in monocytes with the progression of COVID-19 [40, 41].

Since the whole plasmablast cluster had a high attention score, we hypothesized that it might be related to compositional differences. Hence, we next investigated which cell types had the biggest compositional changes between conditions. We found that plasmablast and platelet populations were in the top five (**Fig. 3e**), so MultiMIL identified compositional changes in these two cell types as indicative of disease progression, also reported in [42]. We additionally ran Milo [13] on the same embeddings and found that cell populations identified by MultiMIL, e.g., CD16 monocytes and platelets, were among the cell types with the highest log-fold-change in composition identified by Milo (**Suppl. Fig. 8d**). We note that Milo allows comparisons between two conditions, while MultiMIL identifies condition-specific cell states for multiple classes simultaneously. To examine how dependent the cell attention scores are on the input embedding, we compared cell types with the highest attention scores obtained from MultiMIL embeddings and totalVI embeddings and found that the same cell types were identified (**Suppl. Fig. 8e**).

Finally, we looked at the robustness of cell attention scores. We observed that the scores are mostly consistent across cross-validation runs (**Suppl. Fig. 9a**). The classification formulation yields more stable results than the regression formulation in terms of which cell types belonged to the group of cells with the top 10% attention scores. We therefore suggest that users default to the classification model when analyzing the attention scores. We also note that the cells with the highest attention score were consistently CD14 and CD16 monocytes (**Suppl. Fig. 9b**). We observed this in most cross-validation runs of the classification model with different seeds using MultiMIL embedding or runs using the totalVI embedding. Additionally, we show that by aggregating cells with the highest attention scores, we obtain sample representations most indicative of the disease stages, compared to averaging all cell embeddings or taking a weighted (by attention score) average of the cell embeddings (**Suppl. Fig. 9c,d**, Methods).

We tested the end-to-end training of the model to assess the feasibility of simultaneous learning of the latent representations and the cell attention weights. However, we observed that since there is no clear ground truth on how well the disease and healthy samples should be integrated, it may be challenging to assess if the model over- or under-integrates (**Suppl. Fig. 7a**). We noticed that the accuracy of the prediction on the validation set increases with higher classification coefficients in the loss function up to a certain point but then declines due to overfitting (**Suppl. Fig. 7b**). We therefore recommend that the users train the model in the two-step setting, i.e., first the integration module, then the prediction module. It is also possible to use existing atlases to skip the first step, which will be discussed next.

MultiMIL identifies a subpopulation of IPF-associated macrophages in human lung

Single-cell atlases provide integrated and cell-type-harmonized representations of different systems or organs of interest. These atlases can comprise hundreds of donors, which in turn is crucial to understanding the disease variability and potential therapeutical targets [2]. We demonstrate how MultiMIL can be utilized with existing single-cell atlases. Since MultiMIL's integration and prediction modules can be trained separately, we can train the prediction module directly on the atlas embeddings. The Human lung cell atlas (HLCA) [43] consists of healthy and diseased donors integrated into a common latent space. We investigated idiopathic pulmonary fibrosis (IPF) and compared diseased and healthy samples. To this end, we selected the healthy and IPF individuals from the atlas and trained MultiMIL's prediction module in a 5-fold CV setting (**Fig. 4a**). MultiMIL outperformed other baselines on the prediction task (**Fig. 4b**). We note that other models also achieved high accuracy (>80%). If users are only interested in the binary classification task and not the interpretability aspects, then mean-embedding baselines provide a satisfactory performance (**Fig. 4b**).

We examine the learned cell attention scores to analyze which cell states the model learns to associate

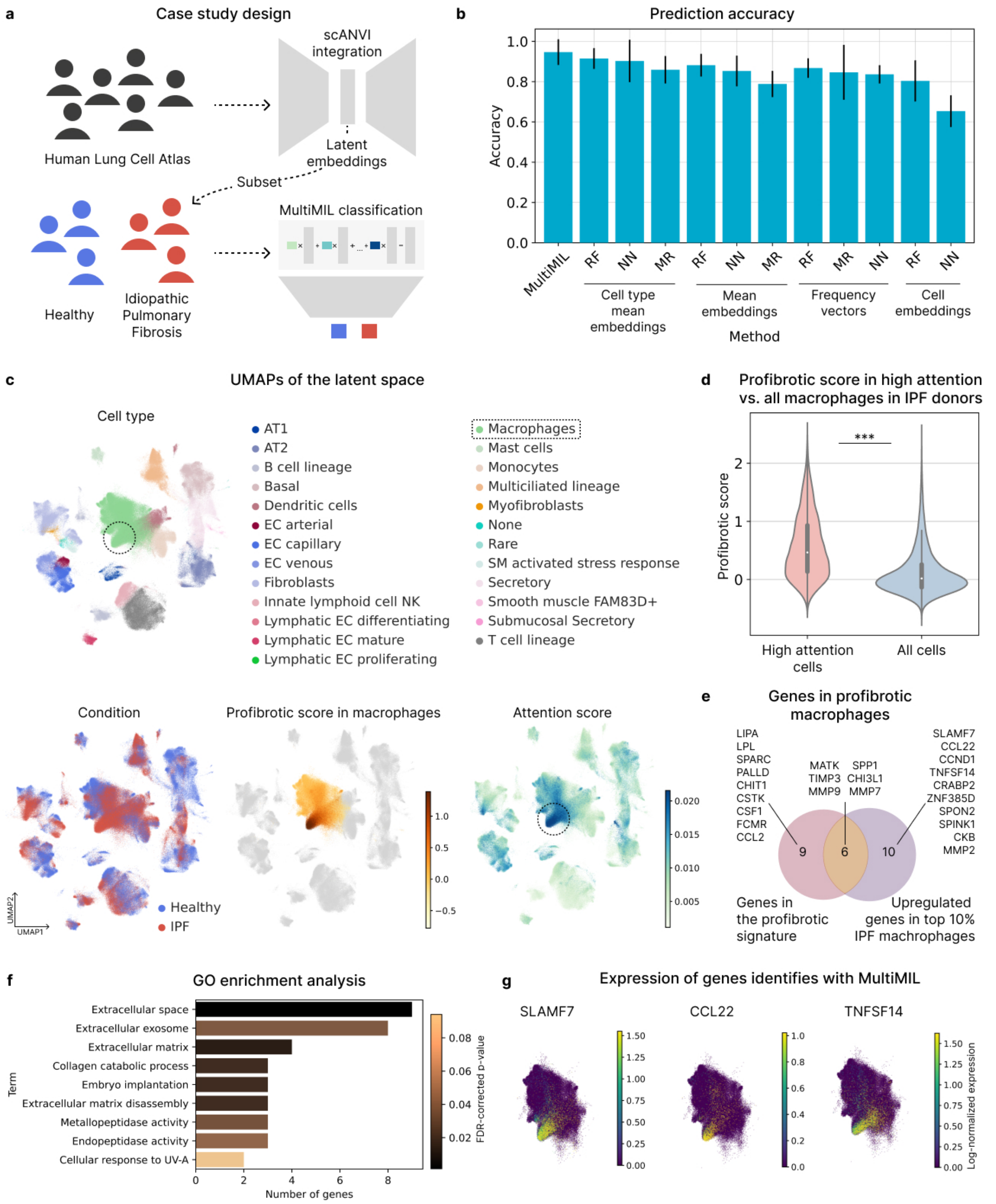


Figure 4: **MultiMIL identifies IPF-associated cell states in human lung macrophages (a)** Case study design. Pre-trained embeddings from the HLCA [43] were subset to healthy and IPF samples and used to train MultiMIL's classification module on the binary classification task. **(b)** A bar plot showing average accuracies and standard deviations of the five cross-validation runs on the prediction task. **(c)** UMAPs of the original latent space from the HLCA colored by cell type (top), condition (bottom left), profibrotic score calculated in macrophages (bottom middle) and cell attention score (bottom right). A subpopulation of macrophages has a high attention score, so we investigate these cells further. **(d)** Violin plots showing the profibrotic score in high-attention macrophages and all macrophages from IPF donors (p-value<0.001, two-sided t-test). **(e)** A Venn diagram with the genes in the profibrotic signature, the number of genes that are upregulated in the high-attention macrophages compared with all macrophages from IPF donors, and the number of genes in the intersection of the two sets. **(f)** GO enrichment analysis of the upregulated genes in the high-attention macrophages. **(f)** UMAPs of the macrophages with the expression of *SLAMF7, CCL22 and TNFSF14*.

with the disease. We notice that a subset of macrophages has the highest scores (**Fig. 4c**), so we first show that MultiMIL identifies a subpopulation of $SPP1^{\mathrm{hi}}$ IPF-specific macrophages [5, 44]. We hypothesize that this subpopulation corresponds to profibrotic macrophage populations reported in previous studies [45, 46]. To confirm, we calculate the profibrotic score based on the profibrotic signature introduced in [45] (**Fig. 4c**). We select macrophages from IPF donors and show that the cells with the highest attention score (top 10%) have a significantly higher profibrotic score than all IPF macrophages (**Fig. 4d**). MultiMIL also identifies a KRT17^{+} subpopulation of basal cells (**Suppl. Fig. 10b**) that previously have been reported to be associated with IPF [5, 47].

Cells with high attention can also be used for novel gene signature discovery or to expand the existing signatures. We demonstrate how to identify the gene signature of the IPF-associated macrophage subpopulation using only the attention scores and not relying on previous knowledge. We ran edgeR [12] to find differentially expressed genes between IPF macrophages with the top 10% highest weight and all IPF macrophages and identified 16 significantly upregulated genes. Comparing these 16 genes with the genes from the profibrotic signature, we find the overlap of 9 (out of 15) genes (**Fig. 4e**).

The genes identified solely from MultiMIL's high attention group include *SLAMF7*, which has been previously reported to regulate the immune response in lung macrophages during polymicrobial sepsis and COVID-19 [48, 49]. Elevated levels of *CCL22* have also been found in patients with IPF [50, 51]. *TNFSF14* promotes fibrosis in the cardiac muscle and atria [52], lung [53] and kidney [54]. Interestingly, *TNFSF14* has been reported to regulate fibrosis in both structural and immune cells [53] (**Fig. 4g**).

IPF is characterized by the excessive accumulation of the extracellular matrix (ECM) and the disrupted balance between ECM production and degradation, where matrix metalloproteinase (MMP) and the tissue inhibitor of metalloproteinase (TIMP) systems play an important role also in macrophages [55]. We found that *TIMP3, MMP7 and MMP9* were reported as part of the profibrotic signature and identified in our DE test. Several other genes that we found, namely, *CCND1, CRABP2, SPON2, SPINK1, CKB and MMP2*, all have been linked to the ECM remodeling [56–61]. We additionally performed Gene Ontology (GO) enrichment analysis [62, 63] on the 16 genes upregulated in the high-attention group and found that the majority of the significantly enriched terms were associated with the ECM (**Fig. 4f**).

Discussion

MultiMIL is a deep-learning-based model for integrating multimodal single-cell data and identifying disease-associated states. It combines cVAE, attention pooling, and multiple-instance learning to provide a comprehensive pipeline for building and analyzing single-cell atlases. Our model integrates paired and partially overlapping single-cell data and uniquely allows for the reference mapping of unimodal and multimodal query samples. We demonstrated that the MIL approach to sample-level classification from single-cell measurements outperforms classical baselines while offering additional interpretability that other models lack. Specifically, MultiMIL can identify transcriptomic and compositional changes driving the disease by analyzing the learned attention scores.

The field of spatial multiomics is rapidly developing [64], and we expect future multimodal models to include spatial data types. Foundation models offer a promising avenue for such endeavors, as some already incorporate multimodal integration as a downstream application [65]. Due to its modular architecture, MultiMIL could be enhanced to work in the spatial domain, enabling the integration of spatial information with other modalities.

Several other MIL models [66, 67] have shown promising results when applied to whole slide images, and initial works in the single-cell field have utilized them in imaging or genomics applications [16, 68]. This work demonstrates the potential applications and advantages of the MIL approach in single-cell multiomics. Future research should benchmark different MIL-based models to identify the most effective strategies for various single-cell applications.

As a deep-learning method, MultiMIL is subject to variability in downstream results due to the stochastic nature of the training process. Additionally, the complexity of the model introduces numerous hyperparameters, necessitating extensive optimization experiments.

We note that new metrics tailored specifically for multimodal integration are required to better assess the quality of the integrated latent space [69]. While some papers on multimodal integration use scIB metrics [70, 71], others provide overviews of metrics explicitly introduced for the multimodal case [72]. Developing and standardizing such metrics will be crucial for future research.

The field of single-cell multiomics is expected to grow rapidly, especially with the ongoing efforts of the Human Cell Atlas (HCA) project [73]. As more large-scale atlases are released, MultiMIL can be readily applied to these datasets to identify cell states potentially relevant to various diseases. This will be particularly impactful in complex diseases such as Alzheimer's, where large cohort datasets are already available [74, 75]. MultiMIL's ability to integrate and analyze these expansive datasets will facilitate the discovery of novel disease-associated cell states and mechanisms.

Additionally, MultiMIL can be utilized for perturbation studies to understand how cells respond to various treatments or environmental changes. This application is crucial for identifying potential therapeutic targets and understanding drug response mechanisms [76]. By analyzing perturbation data, MultiMIL can reveal how different cell states shift in response to specific interventions, providing insights that can guide the development of patient-tailored drugs. This approach not only helps in identifying effective treatments but also in customizing therapies to individual patients based on their unique cellular responses, thereby enhancing the precision and efficacy of medical interventions [77].

MultiMIL offers an innovative approach to linking single-cell-level and sample-level data, identifying biologically meaningful disease-associated cell states. By accommodating multimodal or unimodal data, raw data, or existing atlases, the model provides computational biologists with a versatile tool for various applications.

Code availability

The package is available at `http://github.com/theislab/multimil`. The code to reproduce the results and figures is available at `http://github.com/theislab/multimil_reproducibility`.

Data availability

All datasets analyzed in this manuscript are public and can be downloaded through `http://github.com/theislab/multimil_reproducibility`.

Author Contributions

A.L. and M.L. conceived the project with contributions from F.J.T. A.L. and M.L. designed the algorithm. A.L. implemented the algorithm with contributions from A.O. A.L. performed the paired integration and prediction benchmarks. A.L., M.S. and A.S. ran the trimodal integration and query-to-reference mapping experiments. A.L., S.H.Z. and A.A.M. analyzed cell attention scores for the PBMC case study. A.L. performed the analysis for the HLCA. F.C. curated the datasets and performed label harmonization for the first experiments in the project. All authors contributed to the manuscript. M.L. and F.J.T. supervised the project.

Acknowledgments

We thank Christopher Lance for the help with the cell type harmonization of the NeurIPS 2021 datasets, Lisa Sikkema for the feedback on our analysis of the HLCA, Michaela Müller for patiently answering all the pipeline infrastructure questions, Luke Zappia and Janneke Hulsen for figure feedback, Jan Engelmann and Vladimir Shitov for discussions on MIL and patient representation learning, and Malte D. Luecken for the constructive feedback throughout the project. We thank the `scverse` community (especially the developers and the maintainers of `scanpy`, `muon` and `scvi-tools` packages) and all the Theis lab for valuable discussions.

Competing interests

M.L. consults Santa Anna Bio, owns interests in Relation Therapeutics, and is a scientific co-founder and part-time employee at AIVIVO. F.J.T. consults for Immunai Inc., Singularity Bio B.V., CytoReason Ltd, Cellarity, Curie Bio Operations, LLC and has an ownership interest in Dermagnostix GmbH and Cellarity.

Methods

MultiMIL

MultiMIL is a generative model based on conditional variational autoencoders (cVAEs) [78] with an additional multiple instance learning (MIL) module on the latent space. The architecture consists of three main parts: encoders, decoders and the MIL module. Multimodal single-cell data (together with the batch covariate) is first fed into the encoders, which output parameters of unimodal marginal distributions. Then, a product-of-expert layer calculates the joint distribution parameters from the marginal distributions' parameters. In the latent space, we sample from the joint distribution and then feed the latent embeddings to the decoders (concatenated with batch covariates) and the MIL classifier module. Decoders learn the parameters of the distributions assumed for the input data, and the MIL classifier learns to predict classification labels for a "bag" of cells. In the following, we explain the input to the model in more detail and how each component is trained.

MultiMIL training

We assume there are several single-cell multimodal datasets, each consisting of multiple patients with disease labels available for all patients. Here, we will describe the end-to-end training of the VAE and the MIL modules and discuss the differences between integration-only and prediction-only training later. Single-cell datasets are usually confounded by the technical batch effect, but to simplify the notation, we will treat each dataset as one technical batch. In this section, we will refer to the experimental batches in an experiment or a dataset as "technical batches" or "batch covariates". In contrast, the computational batches, i.e., mini-batches on which machine-learning models are trained, are referred to as "batches" or "training batches".

We denote single-cell datasets as $\{D_1, \ldots, D_k\}$ with corresponding batch covariate labels $\{c_1, \ldots, c_k\}$ and assume that the datasets consist of patients $\{p_1, \ldots, p_d\}$ with corresponding disease labels $\{l_1, \ldots, l_d\}$. We also assume that the datasets are multimodal and have m modalities in total.

We will now focus on a single mini-batch and describe one forward pass of the model. Each training batch consists of single-cell data $\{X_1, \ldots, X_m\}$, the technical batch label $\{c\}$, and the patient disease label $\{l\}$. For simplicity, we assume that only cells from one patient are present in each training batch. Hence, the batch input data matrices $\{X_1, \ldots, X_m\}$ correspond to multimodal data from one patient from m modalities, where some matrices may be all zeros if measurements for the corresponding modality are missing. The number of rows in each matrix X_i equals n, which is the number of cells in the mini-batch, and the number of columns equals the number of features in the original input data of modality i. Note that since the data is paired, the rows in different matrices within one batch always correspond to the same cells.

The data matrices are first fed into the modality-specific encoders $e_1, \dots, e_m$. Each encoder layer consists of a linear layer with dropout, layer normalization and a non-linearity, which can be chosen by the user (with leaky ReLU as default). The output of the encoders are the parameters of $p(z|x_1), \dots, p(z|x_m)$, respectively, which are assumed to be normal. Hence, the output is $(\mu_1, \sigma_1), \dots, (\mu_m, \sigma_m)$, where $\mu_1, \sigma_1, \dots, \mu_m, \sigma_m \in \mathbb{R}^{n \times h}$ and h is the number of latent dimensions and each parameter is learned independently for each latent dimension.

We employ the product-of-expert (PoE) [15, 18] technique to determine the parameters of the joint distribution $p(z|x_1, \dots, x_m)$ from $p(z|x_1), \dots, p(z|x_m)$ for cell j and latent dimension p:

$$\begin{aligned} \mu^{j,p} &= (\mu_0 \sigma_0^{-1} + \sum_{i=1}^{m} M_i \mu_i^{j,p} (\sigma_i^{j,p})^{-1})(\sigma_0^{-1} + \sum_{i=1}^{m} M_i (\sigma_i^{j,p})^{-1})^{-1}, \\ \sigma^{j,p} &= (\sigma_0^{-1} + \sum_{i=1}^{m} M_i (\sigma_i^{j,p})^{-1})^{-1}, \end{aligned} \tag{1}$$

where μ_0 and σ_0 are the parameters of the prior $\mathcal{N}(\mu_0, \sigma_0)$, which in our case is standard normal, so $\mu_0 = 0$ and $\sigma_0 = 1$, and M_i is 1 if modality i is present in this particular batch and 0 otherwise. We obtained the closed form above because we assumed all the distributions to be normal [18]. In the next step, we sample the joint representation $z_{\text{joint}} \sim p(z|x_1, \dots, x_m)$ independently for each latent dimension using the reparametrization trick [79].

During the decoding step, the dataset (i.e., the technical batch) information c is concatenated to z_{joint}, and then the concatenated matrix is fed into each of the modality-specific decoders $d_1, \dots, d_m$. The dataset information c is represented as a learnable embedding in a low-dimensional space. The decoders mirror the encoders' architecture and consist of blocks of a linear layer with dropout, layer normalization and non-linearity.

The latent representation z_{joint} is also fed into the MIL module. The first step here is to aggregate the representations of all cells $z_{\text{joint}} \in \mathbb{R}^{n \times h}$ from the batch (i.e., bag) into a $z_{\text{bag}} \in \mathbb{R}^h$. This bag representation corresponds to a pooled representation of a bag of cells. There are several ways to obtain this pooled representation, e.g., applying *max* or *sum* operators, but we follow [17] and apply attention aggregation:

$$z^p_{\text{bag}} = \sum_{i \in \text{bag}} a^i z^{i,p}_{\text{joint}}, \tag{2}$$

where the joint representation of cell i along latent dimension p is denoted as $z^{i,p}_{\text{joint}} \in \mathbb{R}$. Attention weights $a^i \in \mathbb{R}$ are learned with the gated attention mechanism [17, 80]:

$$a^i = \frac{\exp\left[w^T(\tanh(Vz^i_{\text{joint}}) \odot \text{sigm}(Uz^i_{\text{joint}}))\right]}{\sum_{j \in \text{bag}} \exp\left[w^T(\tanh(Vz^j_{\text{joint}}) \odot \text{sigm}(Uz^j_{\text{joint}}))\right]}, \tag{3}$$

where $w \in \mathbb{R}^q, V \in \mathbb{R}^{q \times h}$ and $U \in \mathbb{R}^{q \times h}$ are learnable weights and q is a hyperparameter known as attention dimension.

After the aggregation, z_{bag} is fed into a classifier network, once again consisting of blocks of a linear layer with dropout, layer normalization and non-linearity. The number of neurons in the last layer equals the number of classes. The classification network predicts the distribution of disease labels for a given bag (i.e., patient). We have now described all of the modules in the model and will discuss the training loss.

MultiMIL can be trained end-to-end, meaning that reconstruction and classification tasks are optimized simultaneously; in this case, we adjust the VAE framework to account for the new classification module. As in standard VAE models, we calculate the reconstruction loss and the Kullback-Leibler (KL) loss with monotonic annealing [20, 21]. For a discussion on VAEs for single-cell data modeling, see [81]. The reconstruction loss is calculated separately for each modality, depending on which distribution is assumed for the input data of this modality (e.g., normal, negative binomial or zero-inflated negative binomial). To obtain the final reconstruction loss, the modality-specific reconstruction losses are summed up:

$$\mathcal{L}_{\text{recon}} = \sum_{i=1}^{m} \lambda_i \mathcal{L}^i_{\text{recon}}, \tag{4}$$

where the weights λ_i are all set to 1 by default, but a weighted sum can be calculated instead. The uneven weighting might be beneficial if the range of loss values differs for different distributions (e.g., if one modality is assumed to follow a Gaussian and another modality – negative binomial distribution). This weighting then ensures that the reconstruction loss for each modality has a similar effect on the overall loss. KL loss is calculated between the assumed prior on the latent space (i.e., standard normal) and the learned joint distribution.

Next, we briefly discuss the maximum mean discrepancy (MMD) loss [22, 23]. We employ MMD loss for two purposes: to ensure that the batches are well integrated, i.e., that joint distributions are similar between batches, and that the unimodal representations follow similar distributions. We are interested in the latter if we want to map unimodal queries onto the multimodal reference. In general, MMD loss measures the distance between two distributions P and Q [22]:

$$\mathrm{MMD}(P, Q) = \mathbb{E}_{a,a'\sim P}[K(a, a')] + \mathbb{E}_{b,b'\sim Q}[K(b, b')] - 2\mathbb{E}_{a\sim P,b\sim Q}[K(a, b)], \tag{5}$$

where a, a' and b, b' are samples drawn from the distributions P and Q, respectively, and K is a kernel function. In the implementation, we use multi-scale radial basis kernels [23] defined as

$$K(a, b, \gamma) = \frac{1}{s}\sum_{i=1}^{s} \tilde{K}(a, b, \gamma_i), \tag{6}$$

where $\tilde{K}(a, b, \gamma_i) = \exp(-\gamma_i||a - b||_2^2)$ is a Gaussian kernel and $s, \gamma = (\gamma_1, \ldots, \gamma_s)$ are hyperparameters.

In our case, the MMD loss is calculated either as the sum over all pairs of batch distributions or as the sum over all pairs of unimodal distributions we want to align. In the first case, MMD loss is calculated between pairs of joint representations $z^1_{\text{joint}}, \ldots, z^k_{\text{joint}}$ coming from different batches $c_1, \ldots, c_k$ as

$$\mathcal{L}^{\text{latent}}_{\text{MMD}} = \sum_{i=1,j>i}^{k} K(z^i_{\text{joint}}, z^j_{\text{joint}}, \gamma). \tag{7}$$

In the second case, we calculate the loss between unimodal marginal representations $z_i \sim p(z|x_i)$ and $z_j \sim p(z|x_j)$ for all $i, j \in \{1, \ldots, m\}$, $i \neq j$ as

$$\mathcal{L}^{\text{marginal}}_{\text{MMD}} = \sum_{i=1,j>i}^{m} K(z_i, z_j, \gamma). \tag{8}$$

The final MMD loss is calculated as

$$\mathcal{L}_{\text{MMD}} = \lambda^{\text{latent}}_{\text{MMD}}\mathcal{L}^{\text{latent}}_{\text{MMD}} + \lambda^{\text{marginal}}_{\text{MMD}}\mathcal{L}^{\text{marginal}}_{\text{MMD}}, \tag{9}$$

where $\lambda^{\text{latent}}_{\text{MMD}}$ and $\lambda^{\text{marginal}}_{\text{MMD}}$ are hyperparameters.

The classification loss is calculated as the cross-entropy loss between one-hot encoded true disease labels and the predicted values of the final layer in the classification network. If the user is interested in modeling the disease classes as a progression, the last layer of the classifier network can be changed to a regression head. In this case, the classification loss is calculated as mean squared error loss. For simplicity, we refer to the regression loss as the classification loss.

The MultiMIL final loss function consists of the VAE loss (which in turn consists of the KL loss and the reconstruction loss), the MMD loss and the classification loss:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}} + \lambda_{\text{MMD}}\mathcal{L}_{\text{MMD}} + \lambda_{\text{class}}\mathcal{L}_{\text{class}}, \tag{10}$$

where $\lambda_{\text{KL}}, \lambda_{\text{MMD}}$ and λ_{class} are hyperparameters.

MultiMIL inference

During test time, we aim to predict the disease class for new patients. For simplicity, we again assume that only cells from one patient are present in the training batch. If needed, we first employ scArches [19] to map new data onto the reference to obtain the latent embeddings. Then, the model needs one forward pass through the MIL module described above. The module aggregates the cell representations into a bag representation, which is then classified using the classification network.

Integration-only training

In the above, we described how to train MultiMIL for simultaneous multimodal integration and patient classification, but the model can also be trained on the integration task alone. The model architecture of the VAE network remains the same in this case, but the MIL module is removed. The model is trained by optimizing the same loss function but without the classification loss. Additionally, cells for each training batch are sampled randomly without considering the patient information. The output of the model is then the joint representation for each cell. These learned latent embeddings can be later used to train the MIL module separately.

Prediction-only training

If the user is interested only in the prediction task and already obtained a low-dimensional integrated representation of the data, MultiMIL can be trained in prediction-only mode. In this case, the embeddings are directly fed into the classifier network and only the classifier is trained.

Integration metrics

To assess the quality of the integration, we used several metrics from the scIB package [33]. Note that scIB metrics were designed for unimodal integration, and not all of them can be easily applied in the multimodal case; hence, we chose the metrics that only require the integrated embedding space as input (and not, e.g., the original unintegrated space). In the following, we briefly discuss two metrics for batch removal and four for biological variance conservation. As in scIB, the final score was calculated as 0.4*batch correction + 0.6*biological conservation. For more details on the metrics and the implementation, see [33].

Batch correction

Graph connectivity measures how well cells from each cell type are connected in a k-nearest neighbor graph. If the connectivity is high, then the batch effect was removed sufficiently. Average silhouette width (ASW) compares average distances within a cluster with distances to other clusters. The resulting score reflects how compact the clustering is. For ASW batch, we expect the batch clusters to be well-mixed together for a high batch correction score.

Biological variance conservation

Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) evaluate how well the clustering is aligned with the ground truth labels, i.e., cell type annotations. ASW label is a modification of ASW batch, where we expect the cell type clusters to be compact and separate from other cell type clusters for a high biological conservation score. Isolated label ASW assesses how well rare cell types are distinguishable from the rest of the data.

Benchmarks

Paired integration

We benchmarked five methods for paired integration (MultiMIL, totalVI [31], multiVI [32], MOFA+ [29] and Seurat v4 [30]) on two CITE-seq datasets (NeurIPS 2021 CITE-seq [26], Hao et al. [27]) and two multiome datasets (NeurIPS 2021 multiome [26], 10x multiome [28]). All methods perform multimodal integration of paired data but employ different approaches. MOFA+ is a linear factor model that decomposes the input data into two low-rank matrices, one representing latent factors (i.e., cell embeddings) and the other representing factor effects. WNN is a graph-based method that outputs a nearest-neighbor graph learned from both modalities. totalVI/multiVI are deep-learning VAE-based methods that model and then fit protein-/chromatin-specific distributions. The output of both models is a latent representation in low-dimensional space. We performed hyperparameter optimization for MultiMIL and then set MultiMIL's default parameters for the integration task based on the best-performing values across all datasets. Other methods were run with their default parameters. We report scIB metrics for all methods. Note that for Seurat v4, we obtained the supervised PCA (sPCA) [82] embeddings from the gene expression and the weighted-nearest neighbor graph to calculate the embedding-based metrics. To find the optimal hyperparameters for MultiMIL, we ran a random grid search for the following parameters and values (with a maximum number of iterations of 100):

Hyperparameter	Description	Default	Range
Batch size	size of the training mini-batch	256	$\{128, 256, 512\}$
Learning rate	learning rate parameter	1e-3	$\{1e-6, 1e-5, 1e-4, 1e-3\}$
KL coefficient	weight of KL loss in the overall loss	1e-5	$\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$
Latent dimension	dimensionality of the latent space	16	$\{8, 16, 32\}$
Conditional dimension	dimensionality of the covariate embedding space	16	$\{8, 16, 32\}$
Number of layers	number of hidden layers in encoders and decoders	1	$\{0, 1, 2\}$
Activation function	non-linearity in the network	LeakyReLU	{LeakyReLU, Tanh}

Table 1 | Hyperparameter grid search for MultiMIL's paired integration.

Mosaic (trimodal) integration

We benchmarked MultiMIL against GLUE [25], multiMAP [34] and scMoMat [35] on the mosaic integration task. We subset the NeurIPS CITE-seq and multiome data to Site1 and Site2 and integrated the two datasets. We ran GLUE using paired and unpaired models. GLUE offers two different models to train, one that considers the pairedness of the data points and one that does not (see Methods); we included both models in our benchmark. MultiMIL and scMoMaT output one embedding per cell, while the rest of the methods output an embedding per cell per modality. To be able to fairly compare the methods, we additionally computed a "joint" representation for each cell as the average of the modality representations for both of the GLUE models and MultiMAP (denoted "avrg.").

Trimodal query-to-reference mapping

Seurat v5 and MultiMIL allow query-to-reference mapping onto the atlases. For Seurat's bridge integration, we first build an RNA-seq-only reference atlas from scRNA-seq measurements from the CITE-seq dataset and snRNA-seq measurements from the multiome dataset using data from Site 1 and Site 2. Then we used one donor (donor 7) from Site 3 as a CITE-seq bridge to map ADT data from Site 4 (donor 9) on top of the RNA-seq reference and the same donor from Site 3 as a multiome bridge to map scATAC-seq data from Site 4 (donor 9) onto the same reference.

For MultiMIL, we mapped unimodal queries, namely scRNA-seq, snRNA-seq and scATAC-seq, and multimodal queries, namely CITE-seq and multiome, on top of the built CITE-multiome reference. We ran a hyperparameter search for MultiMIL for the following parameters and values:

MMD loss type refers to how we calculate the MMD loss: ’latent’ means that $\mathcal{L}^{\text{latent}}_{\text{MMD}} = 1$ and

Hyperparameter	Description	Default	Range
KL coefficient	weight of KL loss in the overall loss	1e-2	$\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$
Integration coefficient	weight of integration MMD loss in the overall loss	4000	$\{1000, 2000, 3000, 4000, 5000, 6000\}$
MMD loss	type of the MMD loss	'marginal'	{'latent', 'marginal'}

Table 2 | Hyperparameter search for MultiMIL's trimodal integration and query-to-reference mapping.

$\mathcal{L}_{\text{MMD}}^{\text{marginal}} = 0$; 'marginal' means that $\mathcal{L}_{\text{MMD}}^{\text{latent}} = 0$ and $\mathcal{L}_{\text{MMD}}^{\text{marginal}} = 1$.

Other hyperparameters were set to their defaults from Table 1. To choose the default parameters, we calculated the scIB metrics on the reference and the mapped queries (with the batch covariate indicating whether the cell came from the reference or the query) to assess the mapping quality.

To assess the accuracy of cell-type transfer, we trained random forest classifiers for each of the query types with `sklearn.ensemble.RandomForestClassifier(class_weight="balanced_subsample")`.

Classification prediction

We compared MultiMIL's predictive ability to several baselines: random forest, multiclass logistic regression, and feed-forward neural networks. We trained each model on the following data input types: mean embeddings, cell type mean embeddings, cell type frequency vectors and cell embeddings. We note that some baselines, namely cell type mean embeddings and cell type frequency vectors, require cell type information, while MultiMIL and the rest of the baselines are entirely unsupervised.

The benchmark was performed on two datasets [38, 43]. HLCA is a unimodal dataset and Stephenson et al. is a CITE-seq dataset. We created 5-fold cross-validation splits based on patient information, i.e., so that cells in each train/validation split come from different patients. We used `sklearn.model_selection.KFold()` to create the splits and `sklearn.metrics.classification_report()` to report the classification accuracy.

We performed a random grid search (with a maximum number of iterations of 100) to find optimal hyperparameters for MultiMIL for each of the datasets and experiments. Table 3 provides the tested parameters.

Hyperparameter	Description	Default	Range
Learning rate	learning rate parameter	depends on the setup	$\{1e-5, 1e-4, 1e-3\}$
Classification coefficient	weight of the classification loss in the overall loss	1.0	$\{0.1, 1, 10, 100\}$
Attention dimension	dimensionality of the attention dimension	16	$\{8, 16, 32\}$
Scoring function	how the attention per cell is calculated	gated attention	{gated attention, attention}
Number of classifier layers	number of hidden layers in the feed-forward classification network	2	{1, 2, 3}

Table 3 | Hyperparameter search for MultiMIL's prediction.

Following the notation from the Results section, attention weights [83] were calculated as

$$a^i = \frac{\exp\left[w^T(\tanh(Vz^i_{\text{joint}}))\right]}{\sum_{k\in\text{ bag}}\exp\left[w^T(\tanh(Vz^k_{\text{joint}}))\right]}, \tag{11}$$

and gated attention weights [84] as

$$a^i = \frac{\exp\left[w^T(\tanh(Vz^i_{\text{joint}}) \odot \text{sigm}(Uz^i_{\text{joint}}))\right]}{\sum_{k\in\text{ bag}}\exp\left[w^T(\tanh(Vz^k_{\text{joint}}) \odot \text{sigm}(Uz^k_{\text{joint}}))\right]}, \tag{12}$$

The batch size was set to 256, the patient batch size to 128 (meaning that in each training mini-batch of size 256, there were two sub-batches of size 128 consisting of cells belonging to one patient each), and the latent and the condition dimensions to 16. Encoders and decoders had one hidden layer each. The default parameters were chosen based on the prediction accuracy of the validation set averaged across five splits.

Next, we discuss the baseline models and the input data in more detail. We performed a hyperparameter grid search for NN-based models and reported the best-performing configuration. Patient disease labels were used as class labels throughout this benchmark apart from the "Cell embedding" input type, where all the cells from a diseased donor were assumed to have the disease class.

Baseline models

- Multiclass logistic regression is an extension to the logistic regression method that allows the prediction of multiple classes. We calculate the probability of belonging to a particular class with a softmax function and calculate the loss as the entropy between predicted probabilities and the true class. We optimize the loss function with gradient descent.
- Random forest was implemented using `sklearn.ensemble.RandomForestClassifier()` with the default parameters.
- Neural network was implemented as a 2-layer feed-forward network with one hidden layer of 64 neurons, batch normalization and ReLU activation. The second linear layer outputs class probabilities. We trained the neural network baselines with Adam optimizer [85] for 200 epochs for sample-level inputs and 30 epochs for cell-level input. Hyperparameter search was run for batch size and learning rate shown in Table 4.

Input data types

- Mean embedding representations were calculated from the latent embeddings with

Hyperparameter	Description	Range
Learning rate	learning rate parameter	$\{1e-5, 1e-4, 1e-3\}$
Batch size for sample-level inputs	size of the training mini-batch	{8, 16, 32, 64}
Batch size for cell-level input	size of the training mini-batch	{128, 256, 512, 1024}

Table 4 | Hyperparameter search NN baseline.

decoupler.get_pseudobulk() specifying the sample parameter and keeping all the cells.

- Cell type-aware mean embedding representations were calculated from the latent embeddings with decoupler.get_pseudobulk() specifying the sample and group (i.e., cell type) parameters and keeping all the cells. To obtain one representation per sample, we concatenated cell type-specific vectors into one vector.
- Frequency vectors were calculated from cell type proportions for each sample.
- Cell embeddings were directly passed to the baselines after integration with MultiMIL, totalVI or published atlases.

Robustness of the integration module

To assess the robustness of the integration, we ran several experiments on the trimodal dataset. We tested several parameters: integration coefficient (i.e., MMD coefficient λ_{MMD}), number of shared features between datasets from different technologies, selection of integration covariates, reference/query ratio and different ways of calculating the MMD loss. Unless the parameter was tested in the experiment, the default parameters used throughout this benchmark were taken from Table 1, and the rest is shown in Table 5.

Hyperparameter	Description	Default	Range
Integration coefficient	weight of the MMD loss in the overall loss	1e4	{1e-3, 1e-2, 1e-1, 1, 10, 1e2, 1e3, 1e4, 1e5, 1e6, 1e7}
Number of shared features	number of shared features between scRNA and snRNA	4000	{100, 500, 1000, 2000, 3000, 4000}
Integration covariate	covariate used for the calculation of MMD	modality	{none, modality, donor}
Batch covariate	covariate(s) used as batch covariate(s)	modality & donor	{modality, donor, modality & donor}
Reference/query split	which sites were used as reference and which as query	Sites 1-3/Site 4	{Sites 1-3/Site 4, Sites 1-2/Sites 3-4, Site 1/Sites 2-4}
MMD type	how MMD loss was calculated	marginal	{marginal, latent}

Table 5 | Parameters tested in the robustness benchmark.

Identification of DA cell states with Milo.

We ran the default Milo [13] analysis on the PBMC dataset using the embeddings learned with MultiMIL. We ran three pairwise analyses comparing mild COVID-19 and healthy, severe COVID-19 and healthy, and severe and mild COVID-19. We show the neighborhoods with spatial false discovery rate (FDR) corrected levels of less than 0.01.

Robustness of attention scores.

To assess the robustness of attention scores, we ran several experiments on the PBMC dataset. First, we ran a 5-fold CV on the same folds, using the same model parameters but changing the random seed using MultiMIL embeddings. Then we also trained the classifier module using totalVI embeddings. To assess the stability of training and attention scores, we looked at the cells with the top 10% attention scores and investigated which cell types they belong to.

We also investigated how well we can predict sample labels with a kNN classifier. We set up a leave-one-out cross-validation experiment using several different aggregation strategies. Sample representations were calculated as a mean of cell embeddings belonging to the sample, mean embedding of cells with top 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% highest attention score and a weighted average of cell embeddings where the weights were the attention weights.

Calculation of the profibrotic signature

To calculate the profibrotic score for macrophages in HLCA, we used the signature from [45]: *SPP1, LIPA, LPL, FDX1, SPARC, MATK, GPC4, PALLD, MMP7, MMP9, CHIT1, CSTK, CHI3L1, CSF1, FCMR, TIMP3, COL22A1, SIGLEC15, CCL2*. The score was calculated with `scanpy.tl.score_genes()`. We performed a two-sided t-test to check for the significance of the score in all IPF macrophages vs. IPF macrophages with the high attention score using `scipy.stats.ttest_ind()`. We used edgeR-QLF [12] to identify the genes differentially expressed in IPF macrophages with the high attention compared to all IPF macrophages and reported genes with a log-fold change greater than 1.5 and FDR-corrected p-value less than 0.01 as up-regulated (see Supplementary Table 1).

Gene Ontology analysis

We used GOATOOLS [86] to run the GO term analysis on the genes that were identified as significantly upregulated in the IPF macrophages with MultiMIL. We followed the tutorial and ran all the functions with their default parameters. We reported the terms with the corrected p-value less than 0.1 as significant.

Datasets

All datasets can be downloaded via `https://github.com/theislab/multimil_reproducibility`.

NeurIPS 2021

The CITE-seq (paired scRNA-seq and ADT) dataset contains 90,261 cells from four sites and 12 batches. The multiome (paired snRNA-seq and scATAC-seq) has 69249 cells from four sites and 13 batches. Both datasets were annotated by the authors and assigned in 30 and 22 cell types, respectively. 'Samplename' was used as the batch covariate.

10x multiome

The data contains 10,000 healthy cells from a multiome experiment. The data does not contain any batches, and the cells are assigned to 11 cell types.

Hao et al.

The CITE-seq data contains 149,926 cells split into two batches. We used the second-level cell type annotations provided by the authors to calculate the scIB metrics. All 228 proteins present in the ADT assay were used in the analyses.

Stephenson et al.

The PBMC dataset contains 647,366 cells from 130 donors, collected at three sites. The ADT panel has 192 proteins. All data points were used for the integration. For the prediction experiment with all COVID-19 stages, we removed non-COVID and non-healthy samples. For the binary experiment, i.e., COVID-19 vs healthy, we subset the data in a balanced way, ensuring that the number of samples from each condition is the same (23).

Sikkema et al.

Human Lung Cell Atlas (HLCA) consists of the core (584,444 cells, 107 donors) and the extension datasets (1,797,714 cells, 380 donors). The core samples are all healthy, while the extension has healthy and diseased samples. In our experiments, we subset the data to healthy and IPF samples in a balanced way, i.e. the number of donors is the same (67) in both groups.

Data preprocessing

For all of the paired experiments, we subset the gene expression datasets to the top 4000 highly variable genes, taking the batch covariate into account with `sc.pp.highly_variable_genes(n_top_genes=4000)` specifying a batch covariate for datasets with batch effects. If the methods required normalized counts as input, we followed standard `scanpy` workflow and applied `sc.pp.normalize_total(target_sum=1e4)` and `sc.pp.log1p()` to the raw counts. ADT counts were central-log-ratio normalized. We selected the top 40000 highly variable peaks for ATAC data with `episcanpy` [87]. To normalize ATAC measurements, we used log-normalization following the `episcanpy` and `muon` tutorials. In the trimodal experiments, we performed the same preprocessing, but subsetting to 20,000 highly variable peaks.

To integrate the PBMC dataset for the prediction experiments, the top 2,000 highly variable genes were selected with `sc.experimental.pp.highly_variable_genes()` using 'Site' as the batch covariate. We preprocessed the ADT data similarly to the above and also removed the isotype controls from the protein matrix.

Running time

We provide training times for the integration module in Table 6, classification module and end-to-end training of models with default architectures. The training was performed on the same GPU server with the following characteristics: Intel(R) Xeon(R) Platinum 8280L CPU with 28 cores, 2.70GHz, Tesla V100-SXM3-32GB GPU. We report the average run time and standard deviation across three runs. We used the PBMC CITE-seq dataset [42], subsetted to healthy, mild and severe COVID-19 in a balanced way, resulting in 256,051 cells. All models were trained for 50 epochs. For the training of the classification module only and the end-to-end training, we modeled the prediction task as either a three-class classification problem or as a regression problem.

	average runtime (s)	standard deviation (s)
integration module	622	2
classification module, classification	356	9
classification module, regression	357	5
end-to-end, classification	937	45
end-to-end, regression	834	89

Table 6

Default architectures

The integration module consists of encoder-decoder pairs, and below we provide the specifications of each pair. Mu and Sigma modules output the μ and σ parameters of the unimodal distributions. Unless specified, the parameters have their default values from PyTorch.

For the model that consists of the integration and the classification networks, the architecture is the same for the integration module, and the default architecture for the classification module is shown below.

We note that we trained the model on the PBMC data with 20 latent dimensions to match the default number of latent dimensions in totalVI for a fair comparison.

Module	Layer

Encoder	Linear(n_input_features, 128)
	LayerNorm
	LeakyReLU
	Dropout(0.2)
	Linear(128, 16)
	LayerNorm
	LeakyReLU
	Dropout(0.2)

Mu	Linear(16, 16)
Sigma	Linear(16, 16)

Decoder	Linear(16 + 16*n_of_covariates, 128)
	LayerNorm
	LeakyReLU
	Dropout(0.2)
	Linear(128, n_input_features)
	LayerNorm
	LeakyReLU
	Dropout(0.2)

Reconstruction decoder	Linear(128, n_input_features) x k, where k depends on the distribution of the input data

Table 7

Module	Layer
Attention aggregator	calculation of attention scores as in Eq. 3 calculation of the weighted sum as in Eq. 2

Classifier	Linear(16, 128)
	Dropout(0.2)
	LayerNorm
	LeakyReLU
	Linear(128, n_classes)

Table 8

Computational resources and package versions

Table 9 provides the version specifications of the main packages used in the benchmarks and the implementation of MultiMIL.

Package	Version	Used in
python	3.10	MultiMIL package
scanpy	1.9.3	pre-processing and MultiMIL package
muon	0.1.5	pre-processing
decoupler	1.4.0	sample-level baselines
torch	2.0.1	neural network baselines and MultiMIL package
sklearn	1.3.0	benchmarks
scib	1.4	benchmarks
scvi-tools	0.20.3	MultiMIL package and paired benchmarks
MOFA+	0.6.7	paired benchmarks
Seurat WNN	4.3.0	paired benchmarks
Seurat Bridge	4.9.9.9058	trimodal benchmarks
scMoMaT	0.2.0	trimodal benchmarks
scglue	0.3.2	trimodal benchmarks
multimap	0.0.1	trimodal benchmarks
R	4.2.2	Seurat and edgeR
edgeR	3.40.0	differential expression testing
snakemake	7.30.1	pipeline to run the classification benchmarks

Table 9

References

[1] Vandereyken, K., Sifrim, A., Thienpont, B. & Voet, T. Methods and applications for single-cell and spatial multi-omics. Nat. Rev. Genet. **24**, 494–515 (2023).

[2] Rood, J. E., Maartens, A., Hupalowska, A., Teichmann, S. A. & Regev, A. Impact of the human cell atlas on medicine. Nat. Med. **28**, 2486–2496 (2022).

[3] Mao, Y. et al. Phenotype prediction from single-cell RNA-seq data using attention-based neural networks. Bioinformatics **40** (2024).

[4] Ravindra, N., Sehanobish, A., Pappalardo, J. L., Hafler, D. A. & van Dijk, D. Disease state prediction from single-cell data using graph attention networks. In Proceedings of the ACM Conference on Health, Inference, and Learning, CHIL '20, 121–130 (Association for Computing Machinery, New York, NY, USA, 2020).

[5] Dann, E. et al. Precise identification of cell states altered in disease using healthy single-cell references. Nat. Genet. (2023).

[6] Zeng, F., Kong, X., Yang, F., Chen, T. & Han, J. scpheno: A deep generative model to integrate scRNA-seq with disease phenotypes and its application on prediction of COVID-19 pneumonia and severe assessment (2022).

[7] Bruggner, R. V., Bodenmiller, B., Dill, D. L., Tibshirani, R. J. & Nolan, G. P. Automated identification of stratifying signatures in cellular subpopulations. Proc. Natl. Acad. Sci. U. S. A. **111**, E2770–7 (2014).

[8] De Donno, C. et al. Population-level integration of single-cell datasets enables multi-scale analysis across samples. Nat. Methods **20**, 1683–1692 (2023).

[9] Boyeau, P. et al. Deep generative modeling for quantifying sample-level heterogeneity in single-cell omics (2022).

[10] Xiong, G., Bekiranov, S. & Zhang, A. ProtoCell4P: An explainable prototype-based neural network for patient classification using single-cell RNA-seq. Bioinformatics (2023).

[11] Skinnider, M. A. et al. Cell type prioritization in single-cell data. Nat. Biotechnol. **39**, 30–34 (2021).

[12] Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edger: a bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics **26**, 139–140 (2010).

[13] Dann, E., Henderson, N. C., Teichmann, S. A., Morgan, M. D. & Marioni, J. C. Differential abundance testing on single-cell data using k-nearest neighbor graphs. Nat. Biotechnol. **40**, 245–253 (2022).

[14] Boyeau, P. et al. Deep generative modeling of sample-level heterogeneity in single-cell genomics. bioRxiv 2022.10.04.510898 (2024).

[15] Hinton, G. E. Training products of experts by minimizing contrastive divergence. Neural Comput. **14**, 1771–1800 (2002).

[16] Sadafi, A. et al. Attention based multiple instance learning for classification of blood cell disorders (2020).

[17] Ilse, M., Tomczak, J. M. & Welling, M. Attention-based deep multiple instance learning (2018).

[18] Lee, C. & van der Schaar, M. A variational information bottleneck approach to multi-omics data integration. In Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, vol. 130 of Proceedings of Machine Learning Research, 1513–1521 (2021).

[19] Lotfollahi, M. et al. Mapping single-cell data to reference atlases by transfer learning. Nat. Biotechnol. **40**, 121–130 (2022).

[20] Kingma, D. P. & Welling, M. Auto-Encoding variational bayes (2013).

[21] Bowman, S. R. et al. Generating sentences from a continuous space (2015).

[22] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. & Smola, A. A kernel Two-Sample test. J. Mach. Learn. Res. **13**, 723–773 (2012).

[23] Lotfollahi, M., Naghipourfar, M., Theis, F. J. & Wolf, F. A. Conditional out-of-distribution generation for unpaired data using transfer VAE. Bioinformatics **36**, i610–i617 (2020).

[24] Argelaguet, R., Cuomo, A. S. E., Stegle, O. & Marioni, J. C. Computational principles and challenges in single-cell data integration. Nat. Biotechnol. **39**, 1202–1215 (2021).

[25] Cao, Z.-J. & Gao, G. Multi-omics single-cell data integration and regulatory inference with graph-linked embedding. Nat. Biotechnol. **40**, 1458–1466 (2022).

[26] Luecken, M. et al. A sandbox for prediction and integration of dna, rna, and proteins in single cells. In Vanschoren, J. & Yeung, S. (eds.) Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks, vol. 1 (Curran, 2021).

[27] Hao, Y. et al. Dictionary learning for integrative, multimodal, and scalable single-cell analysis. bioRxiv (2022).

[28] Datasets - single cell multiome atac + gene exp. - official 10x genomics support. https://support.10xgenomics.com/single-cell-multiome-atac-gex/datasets/2.0.0/pbmc_granulocyte_sorted_10k .

[29] Argelaguet, R. et al. Mofa+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome Biology **21**, 111 (2020).

[30] Hao, Y. et al. Integrated analysis of multimodal single-cell data. Cell **184**, 3573–3587.e29 (2021).

[31] Gayoso, A. et al. Joint probabilistic modeling of single-cell multi-omic data with totalvi. Nature Methods **18**, 272–282 (2021).

[32] Ashuach, T., Gabitto, M. I., Jordan, M. I. & Yosef, N. Multivi: deep generative model for the integration of multi-modal data. bioRxiv (2021).

[33] Luecken, M. D. et al. Benchmarking atlas-level data integration in single-cell genomics. Nat. Methods **19**, 41–50 (2022).

[34] Jain, M. S. et al. MultiMAP: dimensionality reduction and integration of multimodal data. Genome Biol. **22**, 346 (2021).

[35] Zhang, Z. et al. scMoMaT jointly performs single cell mosaic integration and multi-modal bio-marker detection. Nat. Commun. **14**, 384 (2023).

[36] Lance, C. et al. Multimodal single cell data integration challenge: Results and lessons learned **176**, 162–176 (2022).

[37] Hao, Y. et al. Dictionary learning for integrative, multimodal and scalable single-cell analysis. Nat. Biotechnol. **42**, 293–304 (2024).

[38] Stephenson, E. et al. Single-cell multi-omics analysis of the immune response in COVID-19. Nat. Med. **27**, 904–916 (2021).

[39] Sun, D. et al. Identifying phenotype-associated subpopulations by integrating bulk and single-cell sequencing data. Nat. Biotechnol. **40**, 527–538 (2022).

[40] Guo, C. et al. Single-cell analysis of two severe COVID-19 patients reveals a monocyte-associated and tocilizumab-responding cytokine storm. Nat. Commun. **11**, 3924 (2020).

[41] Silvin, A. et al. Elevated calprotectin and abnormal myeloid cell subsets discriminate severe from mild COVID-19. Cell **182**, 1401–1418.e18 (2020).

[42] Stephenson, E. et al. Single-cell multi-omics analysis of the immune response in covid-19. Nature Medicine **27**, 904–916 (2021).

[43] Sikkema, L. et al. An integrated cell atlas of the lung in health and disease. Nat. Med. **29**, 1563–1577 (2023).

[44] Morse, C. et al. Proliferating SPP1/MERTK-expressing macrophages in idiopathic pulmonary fibrosis. Eur. Respir. J. **54** (2019).

[45] Adams, T. S. et al. Single-cell RNA-seq reveals ectopic and aberrant lung-resident cell populations in idiopathic pulmonary fibrosis. Sci Adv **6**, eaba1983 (2020).

[46] Reyfman, P. A. et al. Single-cell transcriptomic analysis of human lung provides insights into the pathobiology of pulmonary fibrosis. Am. J. Respir. Crit. Care Med. **199**, 1517–1536 (2019).

[47] Jaeger, B. et al. Airway basal cells show a dedifferentiated KRT17highPhenotype and promote fibrosis in idiopathic pulmonary fibrosis. Nat. Commun. **13**, 5637 (2022).

[48] Wu, Y. et al. SLAMF7 regulates the inflammatory response in macrophages during polymicrobial sepsis. J. Clin. Invest. **133** (2023).

[49] Simmons, D. P. et al. SLAMF7 engagement superactivates macrophages in acute and chronic inflammation. Sci Immunol **7**, eabf2846 (2022).

[50] Inoue, T. et al. CCL22 and CCL17 in rat radiation pneumonitis and in human idiopathic pulmonary fibrosis. Eur. Respir. J. **24**, 49–56 (2004).

[51] Yogo, Y. et al. Macrophage derived chemokine (CCL22), thymus and activation-regulated chemokine (CCL17), and CCR4 in idiopathic pulmonary fibrosis. Respir. Res. **10**, 80 (2009).

[52] Wu, Y. et al. TNFSF14/LIGHT promotes cardiac fibrosis and atrial fibrillation vulnerability via PI3Kγ/SGK1 pathway-dependent M2 macrophage polarisation. J. Transl. Med. **21**, 544 (2023).

[53] Herro, R. & Croft, M. The control of tissue fibrosis by the inflammatory molecule LIGHT (TNF superfamily member 14). Pharmacol. Res. **104**, 151–155 (2016).

[54] Li, Y. et al. Tumor necrosis factor superfamily 14 is critical for the development of renal fibrosis. Aging **12**, 25469–25486 (2020).

[55] Zhao, X., Chen, J., Sun, H., Zhang, Y. & Zou, D. New insights into fibrosis from the ECM degradation perspective: the macrophage-MMP-ECM interaction. Cell Biosci. **12**, 117 (2022).

[56] Keophiphath, M. et al. Macrophage-secreted factors promote a profibrotic phenotype in human preadipocytes. Mol. Endocrinol. **23**, 11–24 (2009).

[57] Zeng, S. et al. CRABP2 regulates infiltration of cancer-associated fibroblasts and immune response in melanoma. Oncol. Res. **32**, 261–272 (2023).

[58] Zhang, J. et al. The biological functions and related signaling pathways of SPON2. Front. Oncol. **13**, 1323744 (2023).

[59] Xu, L. et al. SPINK1 promotes cell growth and metastasis of lung adenocarcinoma and acts as a novel prognostic biomarker. BMB Rep. **51**, 648–653 (2018).

[60] Loo, J. M. et al. Extracellular metabolic energetics can promote cancer progression. Cell **160**, 393–406 (2015).

[61] Stamenkovic, I. Extracellular matrix remodelling: the role of matrix metalloproteinases. J. Pathol. **200**, 448–464 (2003).

[62] Ashburner, M. et al. Gene ontology: tool for the unification of biology. the gene ontology consortium. Nat. Genet. **25**, 25–29 (2000).

[63] Gene Ontology Consortium et al. The gene ontology knowledgebase in 2023. Genetics **224** (2023).

[64] Baysoy, A., Bai, Z., Satija, R. & Fan, R. The technological landscape and applications of single-cell multi-omics. Nat. Rev. Mol. Cell Biol. 1–19 (2023).

[65] Cui, H. et al. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. Nat. Methods (2024).

[66] Li, B., Li, Y. & Eliceiri, K. W. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. Conf. Comput. Vis. Pattern Recognit. Workshops **2021**, 14318–14328 (2021).

[67] Javed, S. A. et al. Additive MIL: Intrinsically interpretable multiple instance learning for pathology (2022).

[68] Engelmann, J. P., Palma, A., Tomczak, J. M., Theis, F. J. & Casale, F. P. Mixed models with multiple instance learning (2023).

[69] Rautenstrauch, P., Vlot, A. H. C., Saran, S. & Ohler, U. Intricacies of single-cell multi-omics data integration. Trends Genet. **38**, 128–139 (2022).

[70] Brombacher, E., Hackenberg, M., Kreutz, C., Binder, H. & Treppner, M. The performance of deep generative models for learning joint embeddings of single-cell multi-omics data. Front Mol Biosci **9**, 962644 (2022).

[71] Xiao, C., Chen, Y., Meng, Q., Wei, L. & Zhang, X. Benchmarking multi-omics integration algorithms across single-cell RNA and ATAC data. Briefings in Bioinformatics **25**, bbae095 (2024).

[72] Athaya, T., Ripan, R. C., Li, X. & Hu, H. Multimodal deep learning approaches for single-cell multi-omics data integration. Brief. Bioinform. (2023).

[73] Regev, A. et al. The human cell atlas. Elife **6** (2017).

[74] Mathys, H. et al. Single-cell atlas reveals correlates of high cognitive function, dementia, and resilience to alzheimer's disease pathology. Cell **186**, 4365–4385.e27 (2023).

[75] Mathys, H. et al. Single-cell transcriptomic analysis of alzheimer's disease. Nature **570**, 332–337 (2019).

[76] Ji, Y., Lotfollahi, M., Wolf, F. A. & Theis, F. J. Machine learning for perturbational single-cell omics. Cell Syst. **12**, 522–537 (2021).

[77] Gavriilidis, G. I., Vasileiou, V., Orfanou, A., Ishaque, N. & Psomopoulos, F. A mini-review on perturbation modelling across single-cell omic modalities. Comput. Struct. Biotechnol. J. **23**, 1886–1896 (2024).

[78] Sohn, K., Lee, H. & Yan, X. Learning structured output representation using deep conditional generative models. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M. & Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28 (Curran Associates, Inc., 2015).

[79] Kingma, D. P., Salimans, T. & Welling, M. Variational dropout and the local reparameterization trick (2015).

[80] Dauphin, Y. N., Fan, A., Auli, M. & Grangier, D. Language modeling with gated convolutional networks (2017).

[81] Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. Nature Methods **15**, 1053–1058 (2018).

[82] Barshan, E., Ghodsi, A., Azimifar, Z. & Zolghadri Jahromi, M. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. Pattern Recognition **44**, 1357–1371 (2011).

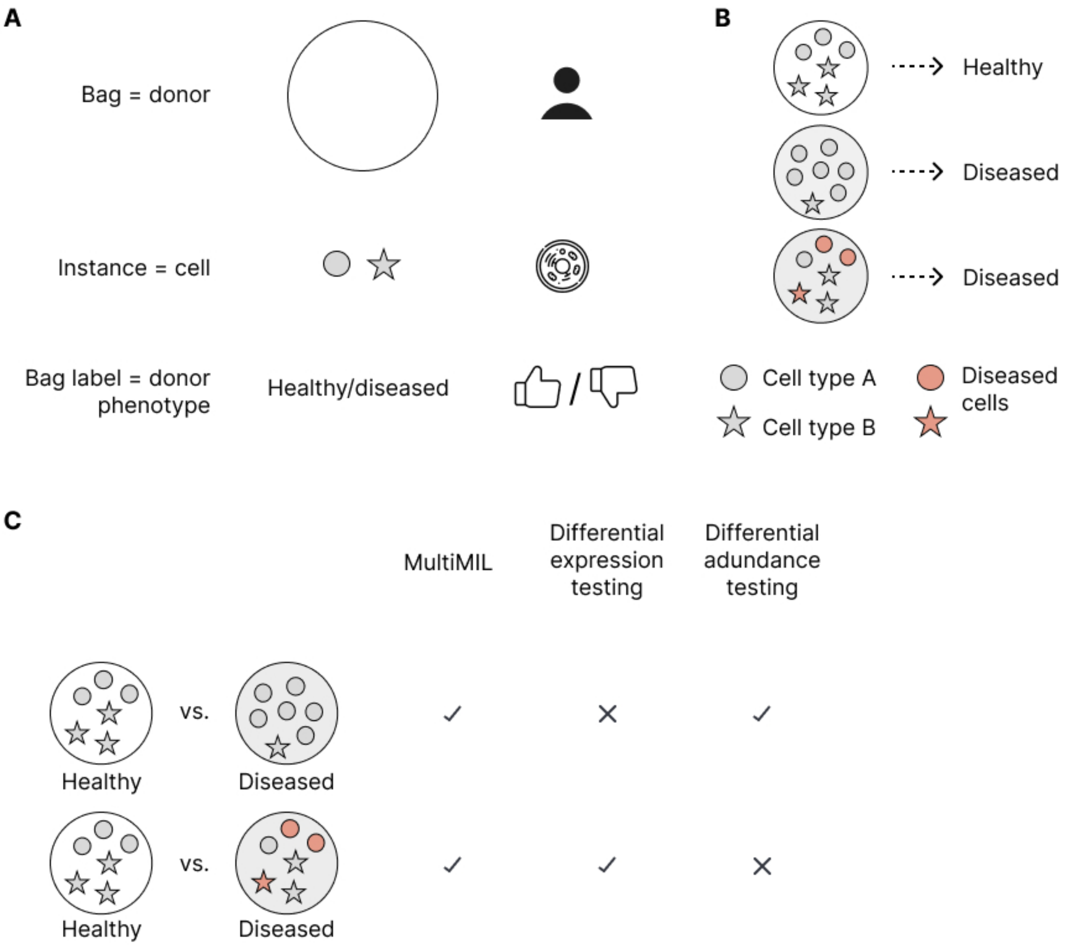
[83] Bahdanau, D., Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate (2014).

[84] Zhang, J. et al. GaAN: Gated attention networks for learning on large and spatiotemporal graphs (2018).

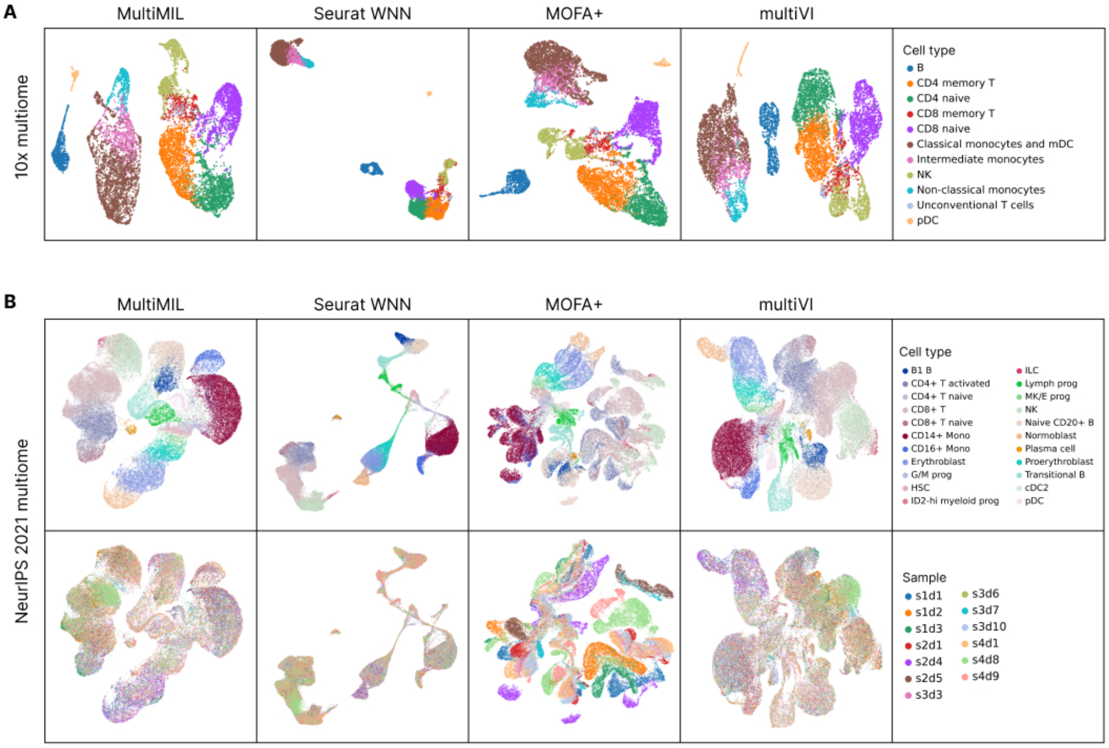
[85] Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization (2014).

[86] Klopfenstein, D. V. et al. GOATOOLS: A python library for gene ontology analyses. Sci. Rep. **8**, 10872 (2018).

[87] Danese, A. et al. EpiScanpy: integrated single-cell epigenomic analysis. Nat. Commun. **12**, 5228 (2021).



Supplementary Figure 1: **Multiple instance learning.** **(a)** In our context, bags correspond to donors, instances to cells and the classification labels are known for bags, i.e., donors. **(b)** Examples of data points in the multiple-instance-learning dataset. Our task is to classify bags into classes and identify cells (i.e. colored instances) that are associated with a certain disease. **(c)** MultiMIL can identify changes in the abundance of cell types between conditions as well as transcriptomic changes.



C

10x multiome

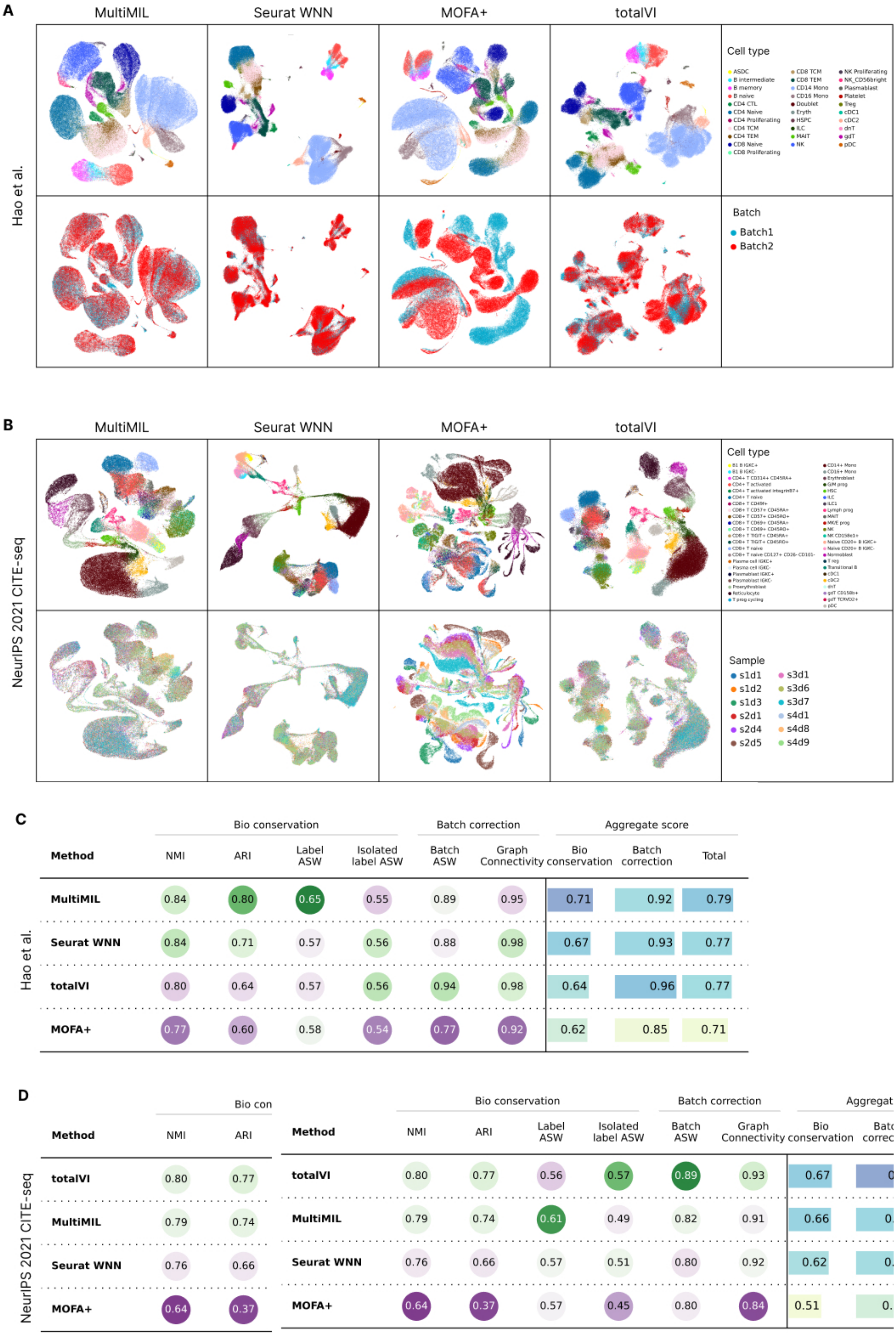
	Bio conservation				Aggregate score	
Method	NMI	ARI	Label ASW	Isolated label ASW	Bio conservation	Total
MultiMIL	0.80	0.74	0.67	0.55	0.69	0.69
Seurat WNN	0.82	0.75	0.61	0.58	0.69	0.69
MOFA+	0.79	0.72	0.63	0.58	0.68	0.68
multiVI	0.78	0.63	0.59	0.55	0.64	0.64

D

NeurIPS 2021 multiome

	Bio conservation				Batch correction		Aggregate score		
Method	NMI	ARI	Label ASW	Isolated label ASW	Batch ASW	Graph Connectivity	Bio conservation	Batch correction	Total
MultiMIL	0.76	0.66	0.60	0.53	0.84	0.95	0.64	0.89	0.74
multiVI	0.75	0.66	0.59	0.51	0.82	0.95	0.63	0.88	0.73
Seurat WNN	0.77	0.67	0.59	0.39	0.81	0.97	0.61	0.89	0.72
MOFA+	0.61	0.26	0.53	0.49	0.81	0.92	0.47	0.87	0.63

Supplementary Figure 2: **Paired integration of multiome datasets.** **(a)** UMAPs of the latent spaces of the 10x multiome dataset, integrated with MultiMIL, Seurat WNN, MOFA+ and multiVI, colored by cell type. **(b)** UMAPs of the latent spaces of the NeurIPS 2021 multiome dataset, integrated with MultiMIL, Seurat WNN, MOFA+ and multiVI, colored by cell type and sample. **(c)** A table showing scIB metric scores for 10x multiome dataset. **(d)** A table showing scIB metric scores for NeurIPS 2021 multiome dataset.

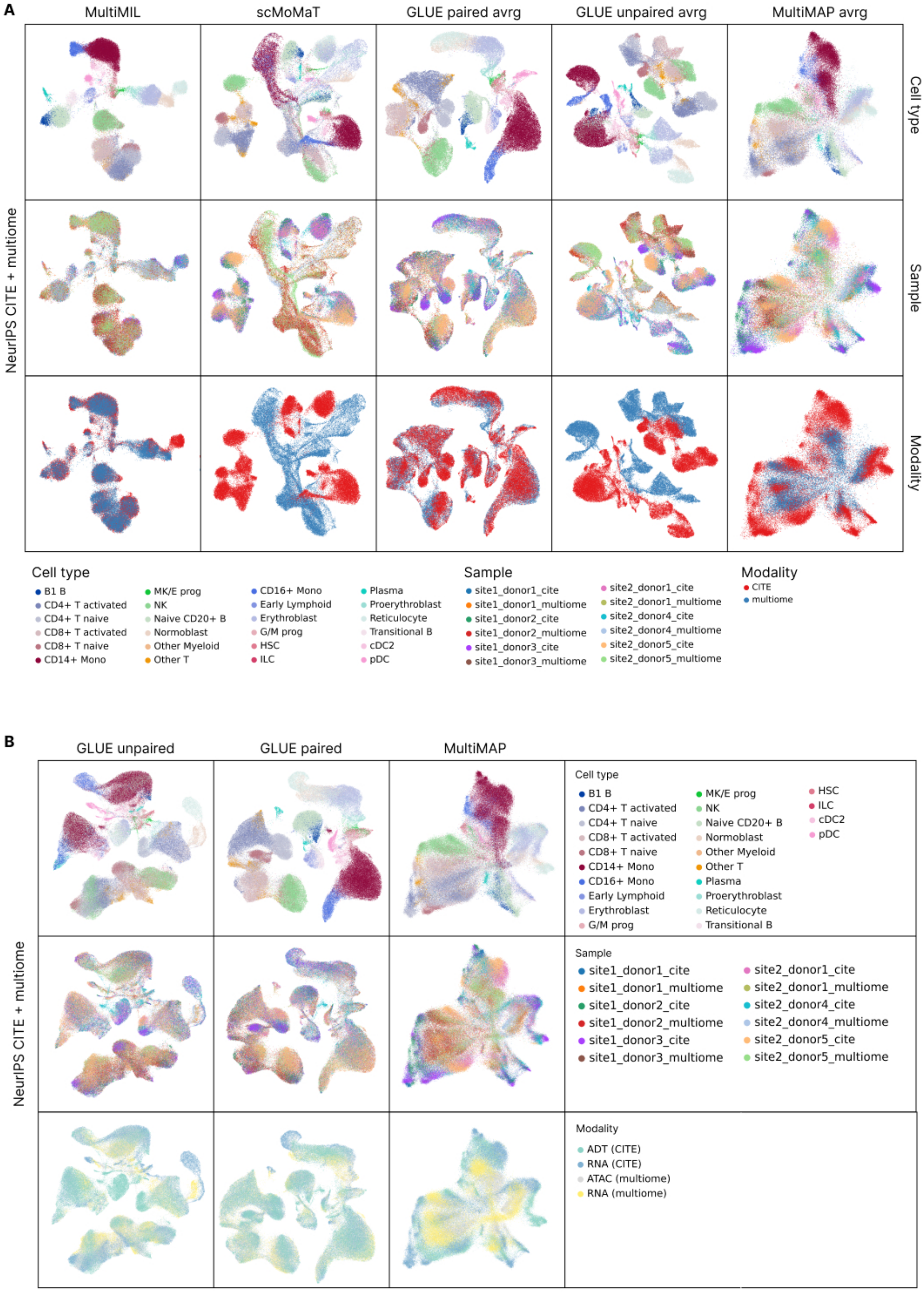


	Bio conservation				Batch correction		Aggregate score		
Method	NMI	ARI	Label ASW	Isolated label ASW	Batch ASW	Graph Connectivity	Bio conservation	Batch correction	Total
MultiMIL	0.84	0.80	0.65	0.55	0.89	0.95	0.71	0.92	0.79
Seurat WNN	0.84	0.71	0.57	0.56	0.88	0.98	0.67	0.93	0.77
totalVI	0.80	0.64	0.57	0.56	0.94	0.98	0.64	0.96	0.77
MOFA+	0.77	0.60	0.58	0.54	0.77	0.92	0.62	0.85	0.71

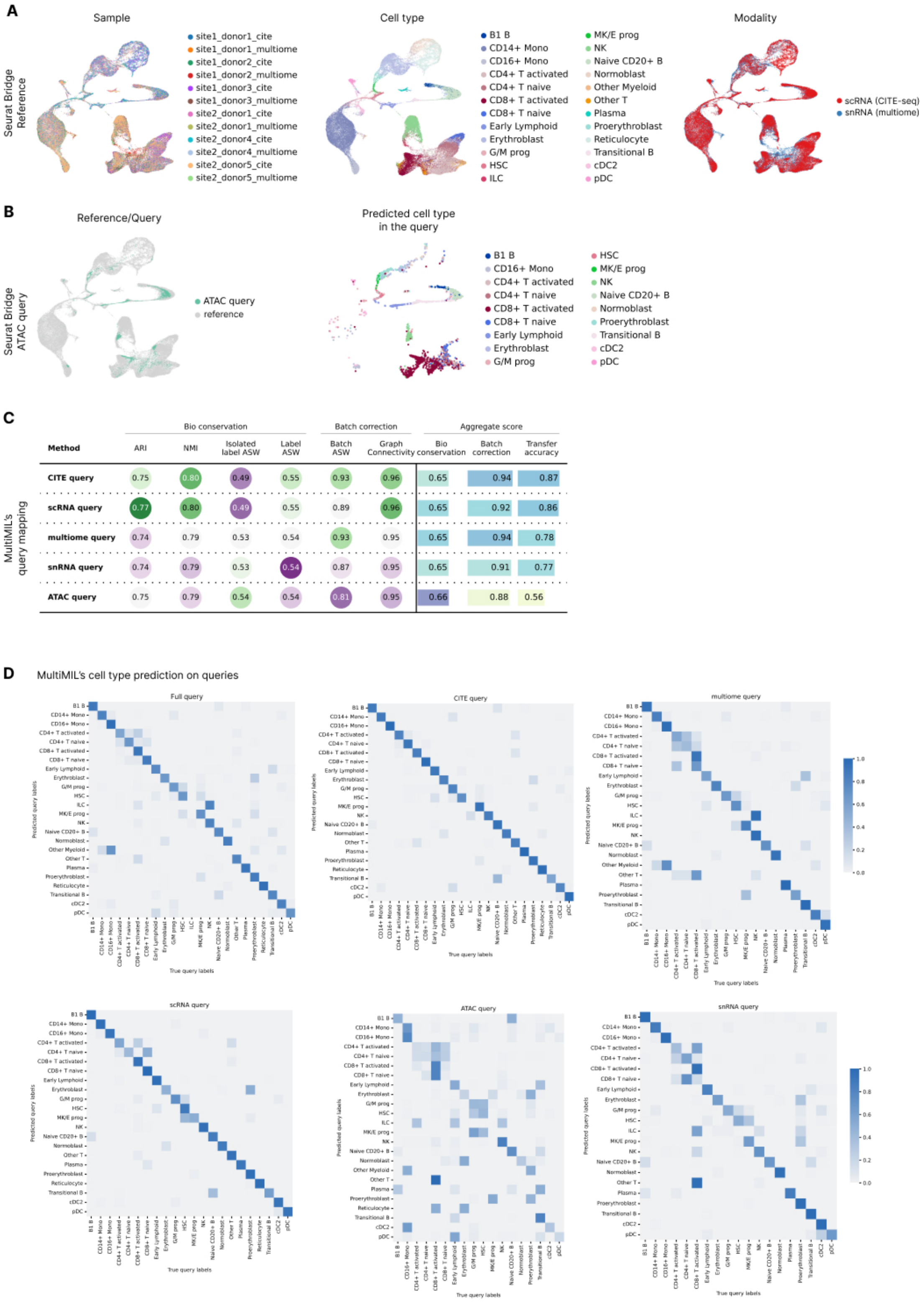
	Bio con	
Method	NMI	ARI
totalVI	0.80	0.77
MultiMIL	0.79	0.74
Seurat WNN	0.76	0.66
MOFA+	0.64	0.37

	Bio conservation				Batch correction		Aggregat	
Method	NMI	ARI	Label ASW	Isolated label ASW	Batch ASW	Graph Connectivity	Bio conservation	Batc correc
totalVI	0.80	0.77	0.56	0.57	0.89	0.93	0.67	0
MultiMIL	0.79	0.74	0.61	0.49	0.82	0.91	0.66	0.
Seurat WNN	0.76	0.66	0.57	0.51	0.80	0.92	0.62	0.
MOFA+	0.64	0.37	0.57	0.45	0.80	0.84	0.51	0.

Supplementary Figure 3: **Paired integration of CITE-seq datasets. (a)** UMAPs of the latent spaces of the Hao *el at.* dataset, integrated with MultiMIL, Seurat WNN, MOFA+ and totalVI, colored by cell type and batch. **(b)** UMAPs of the latent spaces of the NeurIPS 2021 CITE-seq dataset, integrated with MultiMIL, Seurat WNN, MOFA+ and totalVI, colored by cell type and sample. **(c)** A table showing scIB metric scores for Hao *et al.* CITE-seq dataset. **(d)** A table showing scIB metric scores for NeurIPS 2021 CITE-seq dataset.

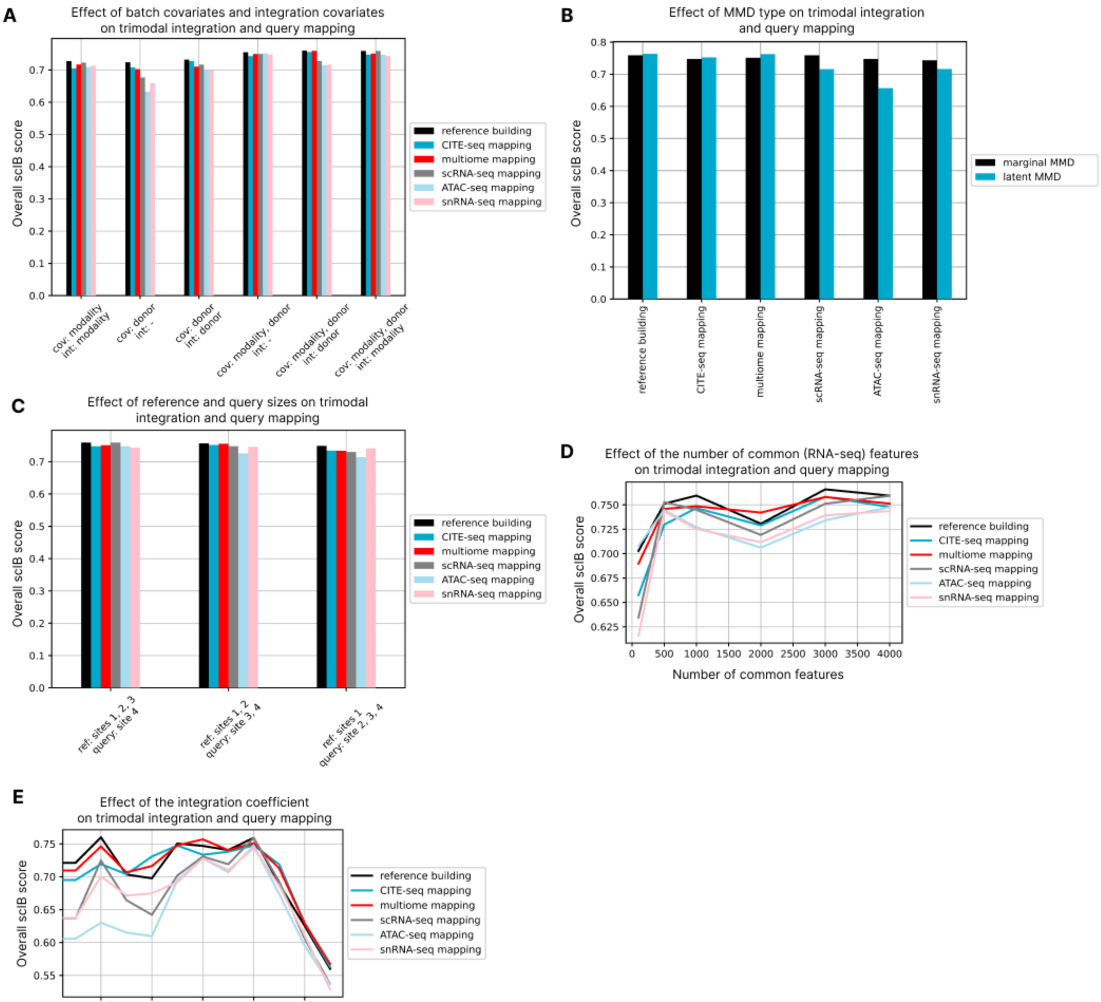


Supplementary Figure 4: **Trimodal reference building.** **(a)** UMAPs of the latent spaces of NeurIPS 2021 multiome and NeurIPS 2021 CITE-seq datasets, integrated with methods that output a representation per cell, i.e., MultiMIL, scMoMaT, GLUE paired (averaged representation), GLUE unpaired (averaged representation) and MultiMAP (averaged representation), colored by cell type, sample and modality. **(b)** UMAPs of the latent spaces of NeurIPS 2021 multiome and NeurIPS 2021 CITE-seq datasets, integrated with methods that output a representation per cell per modality, i.e., GLUE unpaired, GLUE paired and MultiMAP, colored by cell type, sample and modality.

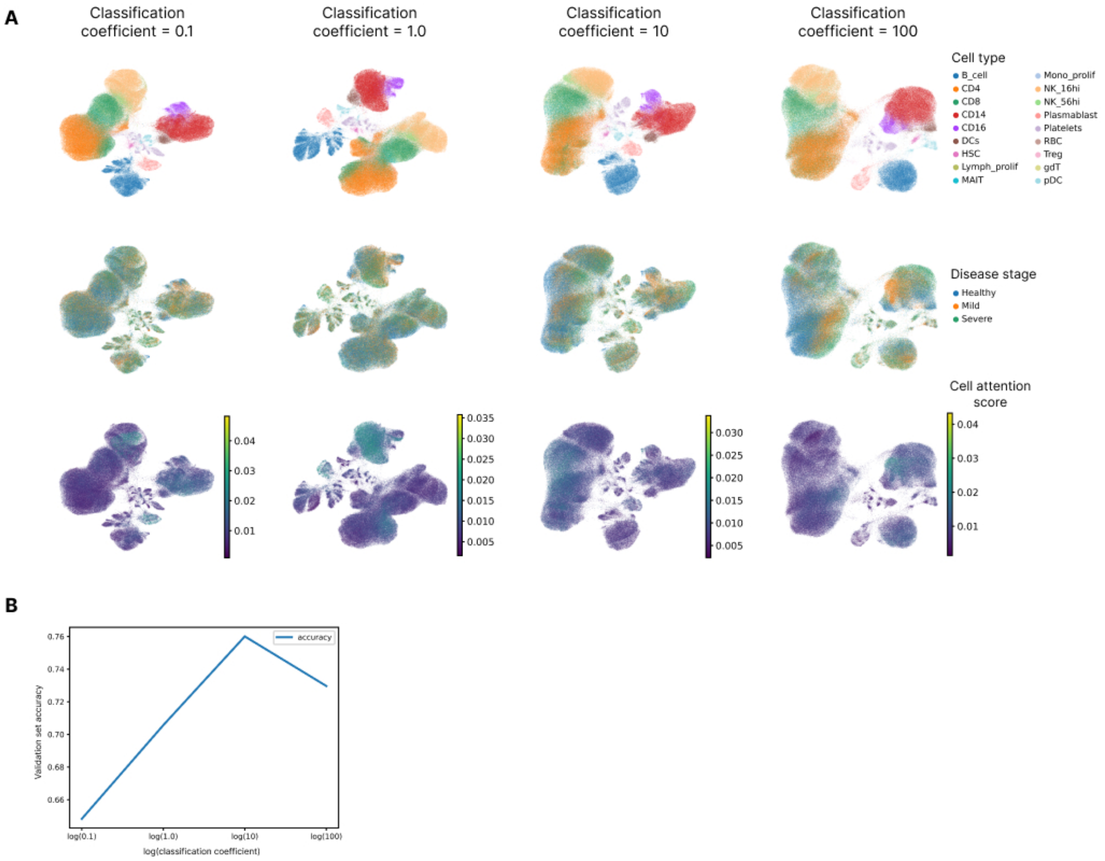


	Bio conservation				Batch correction		Aggregate score		
Method	ARI	NMI	Isolated label ASW	Label ASW	Batch ASW	Graph Connectivity	Bio conservation	Batch correction	Transfer accuracy
CITE query	0.75	0.80	0.49	0.55	0.93	0.96	0.65	0.94	0.87
scRNA query	0.77	0.80	0.49	0.55	0.89	0.96	0.65	0.92	0.86
multiome query	0.74	0.79	0.53	0.54	0.93	0.95	0.65	0.94	0.78
snRNA query	0.74	0.79	0.53	0.54	0.87	0.95	0.65	0.91	0.77
ATAC query	0.75	0.79	0.54	0.54	0.81	0.95	0.66	0.88	0.56

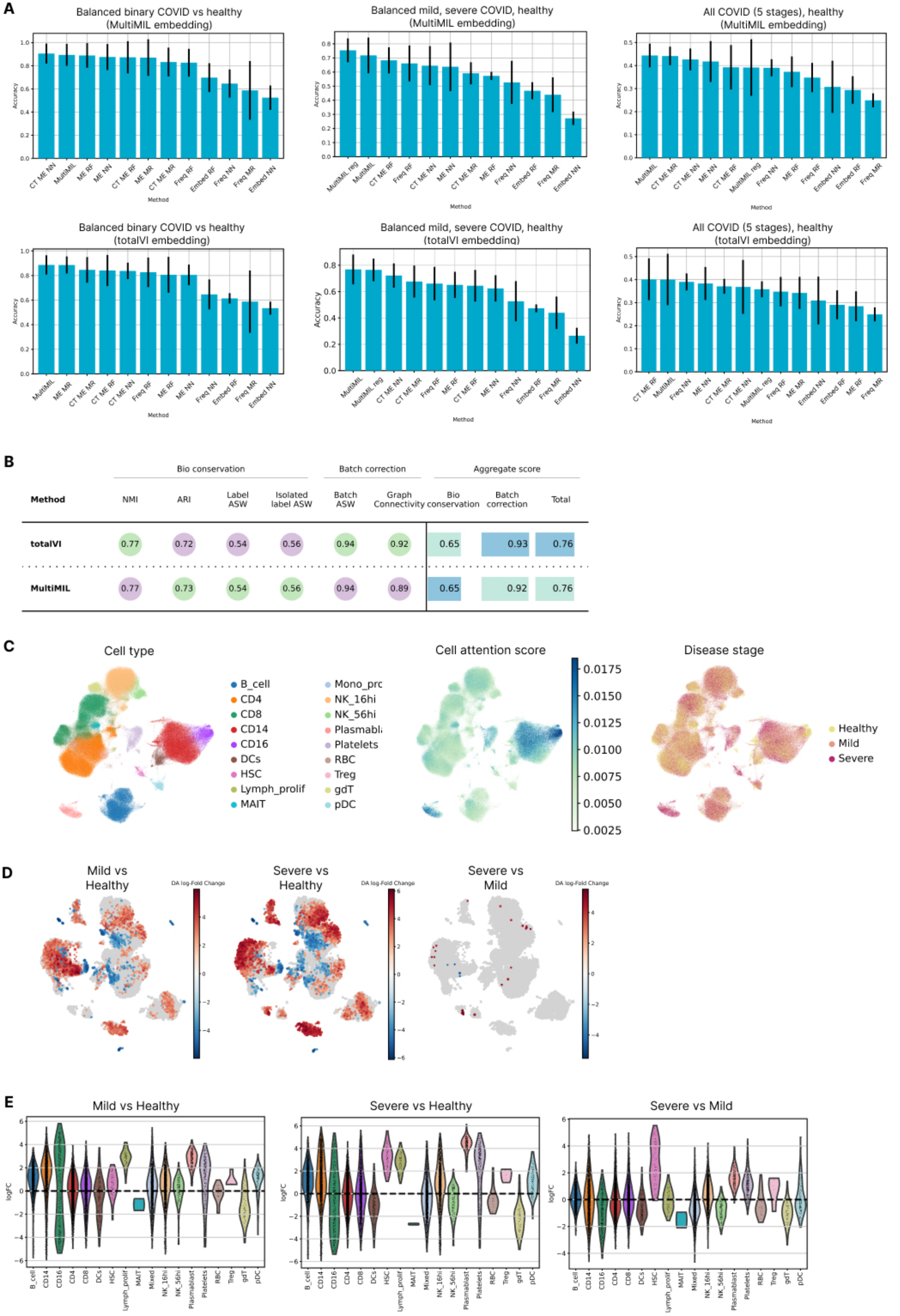
Supplementary Figure 5: **Trimodal query mapping.** **(a)** UMAPs of the integrated scRNA-seq and snRNA-seq from NeurIPS 2021 CITE-seq and NeurIPS 2021 multiome, respectively, with Seurat, colored by sample, cell type and modality/dataset. **(b)** UMAPs of the mapped ATAC query onto the RNA-seq reference with Bridge colored by reference/query and ATAC query only colored by cell type. **(c)** A table with scIB scores calculated for different queries mapped with MultiMIL. **(d)** Confusion matrices between true and predicted (with a random forest model) cell types for the full query and individual queries mapped with MultiMIL.



Supplementary Figure 6: **Robustness of trimodal integration with MultiMIL. (a)** A bar plot showing the effect of batch covariates and integration covariates selection on the scIB overall integration score and query mapping scores. **(b)** A bar plot showing the effect of MMD loss type on the scIB overall integration score and query mapping scores. **(c)** A bar plot showing the effect of the reference and query sizes on the scIB overall integration score and query mapping scores. **(d)** A line plot showing the effect of the number of the common features in the scRNA/snRNA modality on the scIB overall integration score and query mapping scores. **(e)** A line plot showing the effect of the integration coefficient (i.e., the weight of the MMD loss) on the scIB overall integration score and query mapping scores.

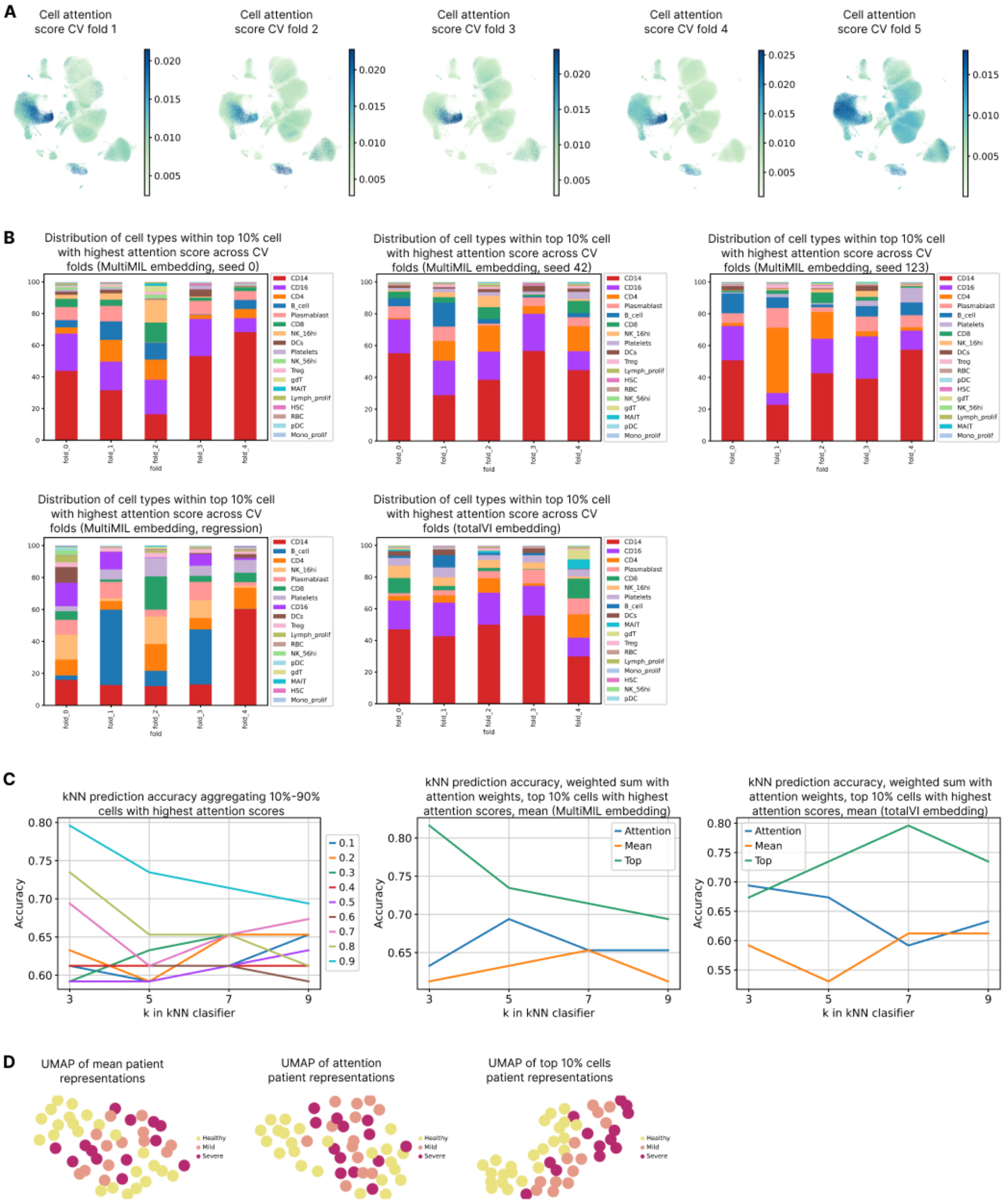


Supplementary Figure 7: **End-to-end training of MultiMIL. (a)** UMAPs of the integrated latent space showing the effect of the classification coefficient colored by cell type (top row), disease stage (middle row) and cell attention (bottom row) for the first CV fold. **(b)** A line plot showing the effect of the classification coefficient on the accuracy of the predicted disease condition on the validation set for the first CV fold.

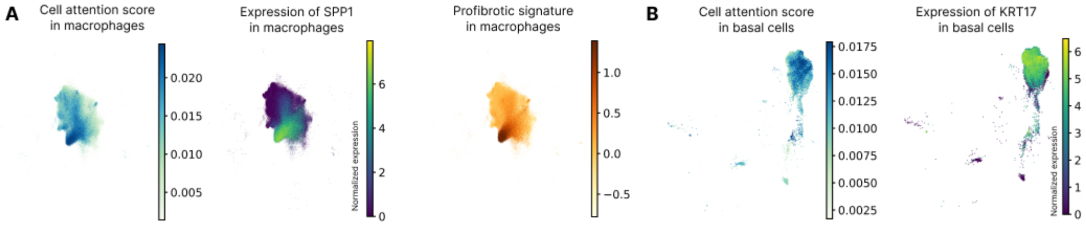


	Bio conservation				Batch correction		Aggregate score		
Method	NMI	ARI	Label ASW	Isolated label ASW	Batch ASW	Graph Connectivity	Bio conservation	Batch correction	Total
totalVI	0.77	0.72	0.54	0.56	0.94	0.92	0.65	0.93	0.76
MultiMIL	0.77	0.73	0.54	0.56	0.94	0.89	0.65	0.92	0.76

Supplementary Figure 8: **Prediction of COVID-19 stages on a CITE-seq PBMC data. (a)** Results of the prediction benchmark on balanced binary (healthy, COVID-19), balanced multiclass (healthy, mild, severe COVID-19) and full data (healthy, 5 COVID-19 stages) using MultiMIL or totalVI embeddings, comparing MultiMIL with the baselines. **(b)** A table showing scIB metric scores comparing MultiMIL and totalVI latent embeddings obtained for the full dataset. **(c)** UMAPs of the totalVI latent space, colores by cell type, cell attention score and disease stage. **(d)** Results of Milo analysis run on MultiMIL's embeddings, mild vs. healthy (left), severe vs. healthy (middle) and severe vs. mild (right), each colored by DA log-fold change (red corresponds to the first condition in the tile). **(e** Violin plots showing DA changes for each of the cell types in mild vs. healthy (left), severe vs. healthy (middle) and severe vs. mild (right).



Supplementary Figure 9: **Robustness of attention scores in PBMC data.** **(a)** UMAPs showing cell attention scores learned in five cross-validation runs. **(b)** Stacked bar plots showing the distribution of cell types with top 10% highest attention scores across five cross-validation runs, comparing runs with different seeds, different MultiMIL setups (classification or regression), and the model ran using totalVI embeddings. **(c)** Line plots showing how well the kNN classifier can predict sample labels from $3, 5, 7, 9$ nearest neighbors when the sample representation was obtained by averaging cell embeddings (MultiMIL's) of cells with top 10%-90% highest attention scores (left); by averaging top 10% (Top), all cells (Mean) and calculating a weighted sum of all cells where the weights are attention scores (Attention) using MultiMIL's (middle) and totalVI's (right) cell embeddings. **(d)** UMAPs of sample representations obtained by averaging cell embeddings (left), by calculating a weighted sum of all cells where the weights are attention scores (middle) and by averaging cell embeddings with top 10% attention scores (right), using cell embeddings from MultiMIL, colored by condition.



Supplementary Figure 10: **IPF in HLCA (a)** UMAPs of macrophages, colored by cell attention, expression of SSP1 and profibrotic signature score. **(b)** UMAPs of basal cells, colored by cell attention and expression of KRT17.