scMinerva: an Unsupervised Graph Learning Framework with Label-efficient Fine-tuning for Single-cell Multi-omics Integrative Analysis

Tingyang YU [1,2,3], Yongshuo Zong [4], Yixuan Wang [1], Xuesong Wang [1], and Yu Li [*1,6]

[1]Department of Computer Science and Engineering, CUHK, Hong Kong SAR, China
[2]Department of Mathematics, CUHK, Hong Kong SAR, China
[3]Department of Information Engineering, CUHK, Hong Kong SAR, China
[4]School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom
[6]The CUHK Shenzhen Research Institute, Hi-Tech Park, Nanshan, Shenzhen, China

Abstract:
Single-cell multi-omics measuring techniques, which simultaneously measures multiple biological contents, such as the epigenome, genome, and transcriptome, are exciting frontiers of exploration in biomedicine. However, integrated analyzing and predicting the cellular states based on these complex multi-omics data are challenging due to the data sparsity, high noise, and computational overhead. To address these challenges, we developed scMinerva, an unsupervised framework for single-cell multi-omics integrated analysis that can effectively classify cell types and stages. We formulate the problem as a heterogeneous graph learning problem and propose a novel biased random walk algorithm omics2vec. It successfully outperforms fully supervised methods on several simulated and real-world datasets while fine-tuned by very few labels. Additionally, scMinerva demonstrates strong label efficiency, is robust to fluctuation in data quality, and allow one omics to compensate weakness in others. Furthermore, we showcase scMinerva's ability to accurately provide prospective biomarkers and predict cell differentiation trends for COVID-19, through the joint analysis of multi-omics data.

1 Background

Single-cell technologies have revolutionized our understanding of biological systems by revealing the significant heterogeneity across different cell types and states. The simultaneous measurement of multiple omics, such as the epigenome, genome, and transcriptome, represents an exciting frontier for single-cell analysis [1, 2, 3]. Consequently, integrated analysis methods are demanded to handle complex multi-omics data and predict cellular states based on them. This capability is crucial for advancing our understanding of disease pathogenesis, identifying new therapeutic targets, and developing personalized treatments [4]. Overall, this trend in single-cell technologies greatly enhances biological discoveries and allows for a more comprehensive understanding of molecular mechanisms.

Despite the benefits of analyzing single-cell multi-omics data, the inherent characteristics of the data make it challenging to analyze computationally. One major challenge is the sparsity of single-cell data, which is due to the lack of gene activity in specific cell types and the relatively shallow sequencing of some droplet-based technologies [5]. For example, when analyzing scRNA-seq data, some genes may be expressed only in a small number of cells, leading to severe dropout in the data matrix. Additionally, the data is often highly noisy due to technical limitations such as amplification bias and low capture rate, which can lead to missing data [6]. Furthermore, the high-dimensional features produced by the multiplexing and high throughput of multi-omics data can result in expensive computational overhead, making the analysis computationally challenging. Most importantly, the annotations of single-cell data are limited since annotating single-cell data could be very labor-intensive and require expert knowledge, and inadequate samples could be given for some rare cases or conditions, making it difficult to study.

*Corresponding Author. Email: liyu@cse.cuhk.edu.hk

To address these challenges, many integrated analysis methods have been proposed for single-cell data, which can be broadly classified into three categories:

1) Latent-space inference methods, which assume a common latent space or kernel shared by all the omics and optimize it with matrix factorization [7, 8, 9] or manifold alignment [9, 10]. These methods are useful when different omics have some shared latent distributions but this is a strict condition for complex multi-omics data and they usually require a more careful data preprocessing.

2) Correlation-based methods that focus on the (dis)similarity measures that correlate different components to each other, including Seurat 4.0 [11], CiteFuse [12], Conos [13], scREG [14], etc. These methods are useful when the omics are not highly correlated with each other but may be sensitive to noise in the data.

3) Deep learning-based methods that learn a model in an end-to-end manner, such as unsupervised methods TotalVI [10] and DeepMAPS [15], as well as semi-supervised method scJoint [16]. These methods can handle complex relationships between the different omics and can be more robust to noise, but they typically require large amounts of annotated data or pre-training, which is a challenging constraint for expensive single-cell annotation.

To summarize, existing methods still have several limitations: 1) Inability to process datasets with more than two omics (such as triple-omics). Most of the existing methods are only designed for datasets with two omics and cannot fully utilize the additional omics, e.g., Seurat 4.0 [11], scREG [14]. Forcibly integrating other omics will potentially impair performance. 2) Sensitivity to noise, especially for the statistical methods. The performance of some existing methods may decrease when there is high noise in certain omics, which requires careful data pre-processing or may lead to failure analysis. 3) Large annotated data requirement for integrated analysis, especially for deep learning-based methods. Most deep-learning-based methods require a large amount of annotated data or pre-training while the annotation process of single-cell data is extremely labor-intensive and requires expert knowledge. With much less annotated data, deep learning methods can not achieve their full potential [17, 18, 19]. 4) Greatly dependent on the clustering results. For previous unsupervised methods, they finished cell-type annotation task by clustering the cells first, and then using the aggregated cluster level expression profiles and the marker genes to label each cluster. Such methods are greatly dependent on the clustering results, which are insufficient for accurate annotation or fine granularity cases.

Recent studies on cell-type annotation method for single-cell RNA sequencing data showed that even a small number of labeled cells (e.g. 10-20% randomly chosen from the sample pool) can be sufficient to train machine learning models for accurate classification, especially when combined with unsupervised methods for feature extraction and clustering [20, 21]. However, this is still a blank area for the label-efficient fine-tuning on single-cell integrated analysis.

Inspired from the recent label-efficiency research and to address these limitations, we propose **scMinerva**, an accurate, robust, label-efficient, and interpretable method that is unsupervised and can flexibly handle any number of omics. It leverages graph convolutional networks (GCNs) and a novel random walk strategy called "omics2vec". Our approach formulates the integration problem as a graph learning problem, constructing a heterogeneous graph by viewing each sample cell as a node and building sub-graphs for each omics. The model then learns a unified embedding for the sample cells by training on this constructed graph using GCNs in an unsupervised manner. In addition, we further introduce a new random walk strategy, omics2vec, that learns to embed nodes by defining a biased random walk procedure. This strategy explores the graph nodes' similarity in a way that balances the exploration of local and global network structures, making it well-suited for the biological setting of single-cell integrated analysis. To classify the sample cells and integrated analyze the cellular conditions, we fine-tune a separate nearest neighbor classifier with very few labels (*i.e.*, 5%-20% of the data) and the produced embedding from scMinerva as inputs to predict cell types or stages. Since this classifier is independent of the unsupervised training stage, our embeddings can be fastly adapted to a annotations with different granulalrity and is relatively less influenced by the clustering results.

We extensively evaluate scMinerva on 4 simulated datasets and 6 real-world datasets, comparing it with previous state-of-the-art methods from different categories using 4 different metrics. Our method outperforms previous state-of-the-art methods by an average of $\sim$15% on dataset SNARE, over 20% on GSE128639, and over 30% on simulated quadruple-omics datasets. Furthermore, compared to other methods, our approach is less sensitive to the amount of annotated data and has a stronger anti-noise ability when facing certain noisy omics. By comparing our results with the performance of single-omics, we found that integrated analysis by scMinerva is robust and always better than any single omics on classification. We showcase the analysis of biomarker detection for Peripheral Blood Mononuclear Cells (PBMCs) using scMinerva. We demonstrate that scMinerva can identify meaningful biomarkers and

that the gene activity value on these biomarkers can expressively reveal the cell stage in a fine-grained manner. Furthermore, we use the predicted results of scMinerva to analyze the cell differentiation trend at the single-cell level for COVID-19-infected T cells. We find that, for COVID-19 infected T cells, their differentiation to $CD4^+$ T_{CM}, $CD4^+$ T_{FH}, $CD4^+$ prolif are activated. In summary, our contributions are as follows:

- Introducing scMinerva which is an interpretable, unsupervised method for integrated analysis of single-cell multi-omics datasets. Our method outperforms previous representative methods and can integrate any number of omics types using a simple-and-independent fine-tuning scheme.
- We extensively demonstrate its excellent performance with data efficiency when fine-tuning with very few annotated data, anti-noise ability when facing certain noisy omics, and robustness that can mostly capture valid information and benefit from different single omics.
- To prove its versatility, we showcase the analysis of biomarker detection and single-cell differentiation trends using the results from scMinerva. With the help of scMinerva, researchers can effectively identify meaningful biomarkers, analyze single-cell differentiation, *etc.*

2 Results

2.1 scMinerva is designed for single-cell multi-omics integrated analysis

In this section, we present *scMinerva*, a novel unsupervised single-cell multi-omics integrated analysis method that employs GCN on heterogeneous graphs and leverages a biased random walk algorithm *omics2vec* for this biological problem, as shown in Figure 1.

Single-cell multi-omics data contains gene activity values for each cell from various omics, each of which measures different biological processes. We create a sub-graph for each omics, each cell will have a mapping node on each sub-graph, which may hold complementary information. ScMinerva is designed to generate latent embeddings for sample cells by connecting the mapping nodes of the sample cell in all the sub-graphs and integrating the information from their neighbors across all omics. There are three main parts of using scMinerva.

The first is to build a heterogeneous graph. After preprocessing the data, we first constructed separate weighted K-nearest neighbor (KNN) graphs for each omics, which we refer to as sub-graphs. Next, we created a heterogeneous graph by linking the mapping nodes of a sample cell in one sub-graph with its corresponding mapping nodes in the other sub-graphs. The next is the omics2vec algorithm, which is inspired by node2vec [22]. Intuitively, it operates as if there is an "explorer" randomly walking on the graph who is gathering information from the neighbors of a sample cell's mapping node. If the explorer is attempting to learn the neighbors' information in one omics, scMinerva allows the explorer to "jump" to the mapping nodes of the cell in other omics, where more information may be available, to gain a deeper understanding of the cell type or stage. Briefly, omics2vec learns the topology of the graph (*i.e.*, edge list) through random walk [23] and incorporates three parameters (p, q, z) in the algorithm. Parameter p determines the likelihood of immediately revisiting a node, q distinguishes between inward and outward nodes corresponding to the former node in the search, and z enables inter-omics transitions within the framework. Omics2vec enables inter-omics transition and can work together with the later GCN model training, which optimizes inter-omics edge weights. Omics2vec has two types: embed for nodes and embed for samples (single cells). The difference is that, after generating walks on the heterogeneous graph, *omics2vec for samples* will map the nodes' indices to their samples' indices to obtain the walks for samples instead of straightly inputting the walks for nodes into the word2vec model. For the details please refer to Section 4.2.4 Omics2vec. The third is the graph convolutional network (GCN) training stage. After obtaining the node embeddings, we fed both the embeddings and the topology of the heterogeneous graph into a GCN model and train it with DeepCluster loss to further optimize the embeddings[24]. The loss will disperse dis-similar nodes on the heterogeneous graph while gathering similar nodes. See more details in the Method part. These three parts complete the unsupervised training stage of scMinerva. With the output sample embedding from scMinerva, we independently fine-tune the model with a nearest-neighbor classifier to fit the sample embeddings and predict their cell types or stages.

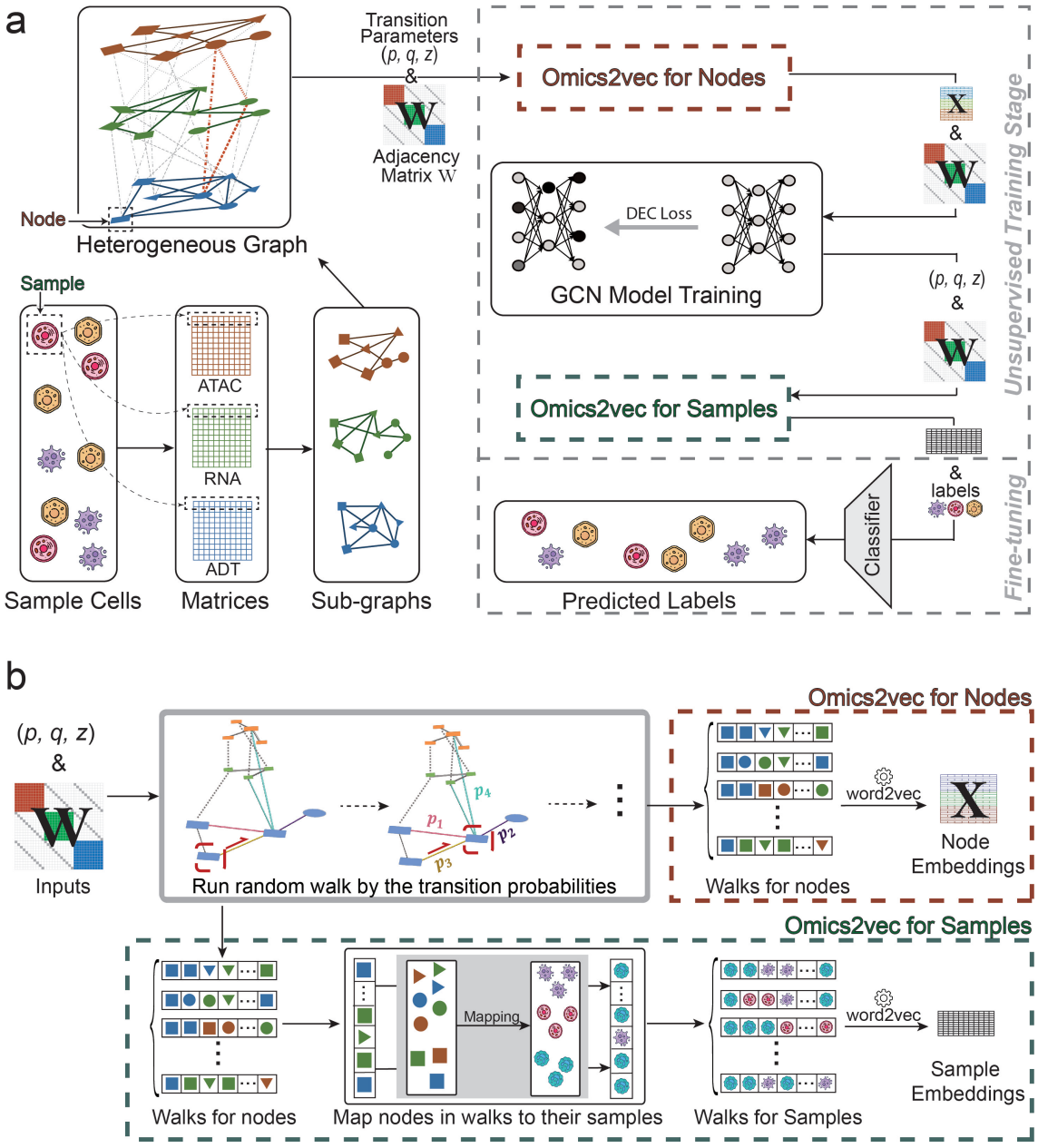

Figure 1: scMinerva workflow and the *omics2vec* algorithm. **a.** scMinerva takes the preprocessed single-cell gene activity matrices as inputs, then constructs a heterogeneous graph, and performs integrated analysis via an algorithm called omics2vec and a graph convolutional network (GCN) model. **b.** Omics2vec has two types. Both have the transition parameters and the adjacency matrix $\mathbf{W}$ as the input, and conduct a biased random walk to explore the graph nodes' similarity. The first is to embed the nodes on the heterogeneous graph: it straightly inputs the walks for nodes into the word2vec model which outputs node embeddings. The second is to embed the samples and completes the integrative analysis: it maps the nodes' indices in the walks to the samples' indices and inputs these walks for samples into the word2vec model which outputs sample embeddings.

2.2 scMinerva outperforms previous state-of-the-art methods in single-cell integrated classification

First, we evaluate the anti-noise ability of our method by simulated datasets in Section 2.2.1 and its accuracy by real-world datasets in Section 2.2.2. To recall, previous unsupervised methods finished cell-type annotation task by clustering the cells first, and then using the aggregated cluster-level expression profiles and the marker genes to label each cluster. Such methods are greatly dependent on the clustering

results, which are insufficient for accurate annotation. Different from them, we evaluated our method in a more label-efficient manner with a simple nearest-neighbor classifier which is independent of the unsupervised training stage. In detail, for the produced embeddings of scMinerva and other methods, we fit them with independent nearest neighbor-based classifiers and fine-tune them using only 10% annotations of the whole training set. This could eliminate the peformance dependence on the results of clustering and fastly adapt the emeddings to annotations with different granularity. For more details on the experimental settings, please refer to Section 4.2.6.

Table 1: Performance comparison on simulated quadruplet-omics data. The first column gives the number of samples in datasets. Bold indicates the best method and the underline indicates the second-best method. Columns named PO1 to PO4 are the performance of four single-omics. All the results are conducted by a KNN classifier with 10% ground-truth labels as the training set. scMinerva outperforms the second-best method by around *30%* across all the metrics.

Size	Metric	**scMinerva**	MOFA+	Conos	Seurat 4.0	PO1	PO2	PO3	PO4
2k	ACC	**0.911**	0.474	<u>0.627</u>	0.263	0.905	0.229	0.351	0.217
	F1-weighted	**0.913**	0.464	<u>0.616</u>	0.209	0.906	0.199	0.333	0.182
	F1-macro	**0.913**	0.462	<u>0.615</u>	0.211	0.907	0.199	0.332	0.182
	ARI	**0.781**	0.127	<u>0.307</u>	0.004	0.775	0.005	0.060	0.002
5k	ACC	**0.952**	0.547	<u>0.764</u>	0.411	0.976	0.236	0.536	0.239
	F1-weighted	**0.952**	0.548	<u>0.764</u>	0.408	0.976	0.223	0.524	0.224
	F1-macro	**0.952**	0.548	<u>0.656</u>	0.407	0.976	0.224	0.522	0.224
	ARI	**0.913**	0.198	<u>0.449</u>	0.099	0.942	0.007	0.250	0.009
10k	ACC	**0.957**	0.691	<u>0.719</u>	0.220	0.986	0.262	0.399	0.211
	F1-weighted	**0.957**	0.686	<u>0.714</u>	0.165	0.986	0.258	0.389	0.206
	F1-macro	**0.957**	0.686	<u>0.714</u>	0.164	0.986	0.258	0.388	0.206
	ARI	**0.895**	0.410	<u>0.450</u>	0.001	0.966	0.017	0.087	0.000
30k	ACC	**0.968**	<u>0.675</u>	0.447	0.256	0.996	0.275	0.300	0.265
	F1-weighted	**0.968**	<u>0.676</u>	0.446	0.230	0.996	0.272	0.297	0.261
	F1-macro	**0.968**	<u>0.677</u>	0.446	0.230	0.996	0.272	0.297	0.261
	ARI	**0.927**	<u>0.377</u>	0.134	0.007	0.990	0.024	0.024	0.025

2.2.1 Simulated data

It is a common agreement that there is a rapid trend in the development of experimental methods which jointly profile three or more omics types [25]. However, it is harder to find specific common latent features or sample correlations when the omics type number is increasing. For example, triplet-omics datasets will be more challenging to handle compared to couplet-omics datasets as different gene activity matrices (omics) have different distributions and certain omics might be polluted by severe natural or artifact noises. Consequently, computational algorithms with a stronger anti-noise ability for more omics data (*i.e.*, triplet-omics, quadruplet-omics) will be greatly required to adapt to this phenomenon. As a start, we first simulate some pseudo-multi-omics datasets to answer the following question: how would noises affect the performance of the integrated analysis methods while the omics type number increases?

Since there exist real-world couplet-omics or triplet-omics datasets, we leave the evaluation on real-world datasets with two omics or triplet-omics to the next subsection. For this section, we boldly assume simulated quadruplet-omics datasets with four pseudo-omics as an perspective for future measurement technological advances. Our intention is to produce simulated quadruplet-omics datasets, in which different gene activity matrices (omics) have different distributions and certain noisy omics are contained. By doing so, we can approximately evaluate the anti-noise ability of methods while the omics type number is increased. Our simulated data and the labels were produced by a single-cell RNA (scRNA) data simulator, splatter [26], and three convolutional neural networks (CNNs). We took the RNA-seq of GSE156478-CITE [27] as an input to obtain simulated RNA-seq. Even though simulating quadruplet-omics datasets could be very challenging, we hoped to distinguish our simulated datasets from trivial linear regression tasks. Therefore, we assumed that there are some commonly shared latent features for real-life omics and trained CNNs with data from real-world datasets to learn these latent features. The three networks were trained by mapping sci-CAR RNA-seq to its ATAC-seq, GSE156478-CITE RNA-seq

to its ADT data, and GSE156478-ASAP ATAC-seq to its ADT. By inputting the simulated RNA-seq to the above networks, we generated four-omics datasets with 5 classes and sample numbers 2k, 5k, 10k, and 30k, respectively. These simulated datasets contain pseudo-RNA-seq (PO1 shown in table 1), pseudo-ATAC-seq (PO2), pseudo-ADT from RNA-seq (PO3), and pseudo-ADT from ATAC-seq data (PO4). Full details on data simulation can be referred to Method Data Simulation from Real-world Datasets.

We fine-tune a 8-nearest neighbor classifier with 10% label to fit the embeddings outputted by all the methods. The results are listed in Table 1, where we compared the performance with Seurat 4.0 [11], MOFA+ [8], Conos [13], and using only single pseudo-omics. To note, TotalVI [10], CiteFuse [12], and DeepMAPS [15] are not capable of processing more than two omics and are thus not included here. They will be compared in the later section on two omics real-world datasets. We examine the performance with accuracy, F1-weighted score, F1-macro score, and adjusted rand score (ARI). Details on these metrics are in Appendix Evaluation metrics.

Among all the metrics and all the simulated datasets, our method showed around 30% improvements over the second-best method. Moreover, it is reasonable that the performance of scMinerva could not exceed the best performance of the single pseudo-omics PO1. This is because the generating process of simulated datasets is not supervised by complementary biological information. As we have demonstrated before, most of the existing methods (especially for the statistical methods) are strongly impaired by the low-quality omics in datasets and perform terribly. Compared to other representative methods, scMinerva guarantees stronger robustness on anti-noise. It performs stably even though there are more challenging or noisy omics in the datasets.

2.2.2 Real-world datasets

The previous subsection has shown that our method outperforms existing methods under challenging cases, i.e., quadruplet-omics datasets or datasets with certain extremely-noisy omics, we further evaluate scMinerva on real-world datasets. Similarly as the last subsection, we run the integrated classification experiments in a label-efficient manner with a simple nearest-neighbor classifier. In detail, for the produced embeddings of scMinerva and other methods, we fit them all with an independent nearest neighbor-based classifier fine-tuned using only 10% annotations of the whole training set. We conduct our evaluation on CITE-seq (GSE128639 [28], GSE156478-CITE [27], COVID-PBMC [29]), ASAP-seq (GSE156478-ASAP [27]), SNARE-seq [30], and scNMT-seq [31]. The selected datasets are of diverse sample sizes, ranging from 1k to 64k. For the COVID-PBMC dataset, we take the healthy samples and critical symptom samples separately and form the COVID-non-covid dataset and the COVID-critical dataset, respectively. We extract these two datasets for the convenience of the later analysis. All datasets were under standard pre-processing and quality control (details in Method Preprocessing). Before preprocessing, most of these datasets have a more than 90% dropout rate and a shallow measurement depth, which is listed in Figure 3.

The existing methods, DeepMAPS, CiteFuse, totalVI, and Seurat 4.0, were all run with their default settings. For MOFA+ and Conos, the dimension of their embeddings needs to be adjusted to a large dataset. So we set the dimension of their output embeddings to 200 to reserve enough latent information as other methods. Notably, MOFA+ was restricted by memory limitation. We failed to run the originally preprocessed data on it with a 32G RAM when the number of samples is greater than 10k. Thus, we apply PCA on the data with 300 components on MOFA+ for dataset GSE128639. Also, DeepMAPS, CiteFuse, and TotalVI are unsuitable for three omics cases, and therefore they are not listed in the chart for COVID-PBMC, scNMT, and GSE128639. With the generated embeddings, we evaluated the classification performance by fitting a K-nearest Neighbor (KNN) Classifier with the number of neighbors as 30 for datasets containing more than 5k samples, and with the number of neighbors as 8 for datasets smaller than 5k using the sklearn library [32]. The details for data preprocessing are listed in Appendix Preprocessing. For full details on the experimental setting, please refer to Method. The results are shown in Figure 2.

Except for scNMT-seq, which measures the differentiation of mouse embryonic stem cells, most of the existing three omics datasets are built on human peripheral blood mononuclear cells (PBMC). Normally for embryonic stem cells, the ATAC-seq, RNA-seq, and DNA Methylation levels are measured. But for PBMC consisting of lymphocytes (T cells, B cells, NK cells) and monocytes, T cell receptor (TCR) and B cell receptor (TCR) measurements are available as well as a rich expression level on surface protein as shown in Figure 2a.

The algorithm scMinerva has been evaluated on several triplet-omics datasets, including scNMT,

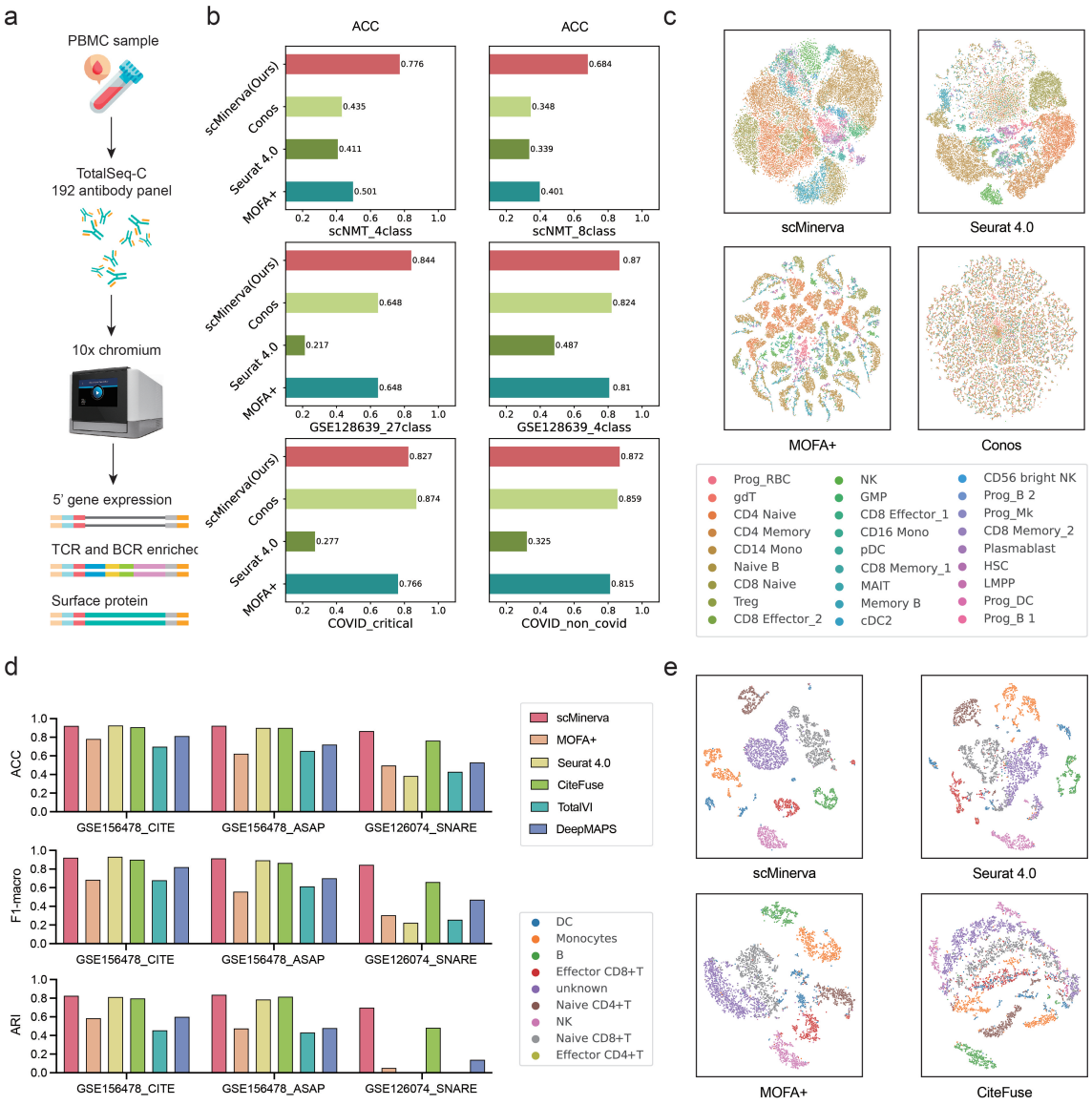

Figure 2: Classification performance on real-world datasets. **a**. The sequencing technique for Peripheral blood mononuclear cells to obtain three-omics data. **b**. Classification accuracy comparison on four triplet-omics datasets with six sets of annotations. We fine-tune the output embeddings with 10% labels. Our method shows a overall-the-best performance. **c**. Visualization on three-omics dataset GSE128639 with 30k cells and 27 classes for scMinerva, Conos, Seurat 4.0, and MOFA+. scMinerva's visualization has the clearest boundary and separates clusters properly. **d**. The classification performance on two-omics datasets for scMineva and five exiting methods. We fine-tune the output embeddings with 10% labels. scMinerva always shows both great and stable ability even on SNARE which has one extremely noisy omics. **e**. Visualization on dataset GSE156478-ASAP with 5k cells and 8 classes. The scatter is colored by ground-truth labels. It is obvious that scMinerva nicely has the most dispersed clustering which best matches the ground-truth clusters.

GSE128639, and COVID-PBMC, with a 10% fine-tuning data of different methods. The results are listed in Figure 2b, which shows that scMinerva outperforms most of the compared integrative analysis methods on classification tasks and has comparable performance with Conos, the best-performance method on the COVID_critical dataset. scMinerva exhibits notable improvements in classification accuracy, with an average increase of 7% and up to 20% on GSE128639 when classifying into 27 classes compared to the second-best method. Furthermore, it demonstrates stability when only 10% labels are used for training. Importantly, different datasets have various versions of data annotations with different numbers of cell types. For example, GSE128639 is annotated by 4 classes and 27 sub-classes. The unsupervised integrated

stage and an independent fine-tuning scheme of scMinerva ensure a faster adaptation to different data annotation granularity. To further evaluate the performance of scMinerva, we visualize the embeddings of the GSE128639 dataset produced by scMinerva, Conos, Seurat 4.0, and MOFA+ via t-SNE in Figure **2c**. GSE128639 contains 30k cells across 27 cell types and cell states. From the scatter plot colored by the ground truth labels, it is evident that scMinerva clusters samples from the same type together and shows clear boundaries between different types. In contrast, MOFA+ shows a dispersed gathering for samples that should be in the same cluster. The embeddings of different cell types in the visualizations of Seurat 4.0 and Conos overlap, indicating that they are not able to effectively discriminate different cell types. These findings strongly support the good overall performance of scMinerva in the classification of triplet-omics datasets.

Moreover, couplet-omics PBMC datasets, GSE156478-CITE and GSE156478-ASAP, were used to test the performance of existing methods for classification. The results in Figure 2**d** show that scMinerva achieved a slightly better performance than the second-best method on these two datasets. Seurat 4.0 and CiteFuse also showed similar performance with scMinerva, but CiteFuse was not adaptive to datasets containing more than two omics, and Seurat 4.0 had poor performance on high-noise datasets such as SNARE. The embedding of GSE156478-ASAP was visualized in Figure 2**e**, and scMinerva was found to have a clearer embedding than the other three methods. However, on the SNARE dataset, scMinerva outperformed all existing methods on classification with an around 15% promotion. This performance gap occurred because scMinerva had excellent anti-noise ability and could effectively handle situations with a severe quality gap between different omics. The result listed below in Figure 3 shows a severe quality gap between two omics of the SNARE dataset, as the performance of ATAC-seq was very poor. However, scMinerva was not affected by the poor quality of certain omics and showed strong robustness. On the other hand, all other methods, including TotalVI, DeepMAPS, Seurat 4.0, and MOFA+, tended to mess up all samples to one class, while CiteFuse was strongly encumbered by the high noise in ATAC-seq. scMinerva achieved an accuracy of over 80% and had a 20% improvement on ARI over other methods.

In conclusion, scMinerva demonstrated good overall performance compared to existing methods in multi-omics data integration. Its anti-noise ability and strong robustness make it a promising method for handling situations with a severe quality gap between different omics.

2.3 Integration on multi-omics data is robust and overall better than any single omics on classification

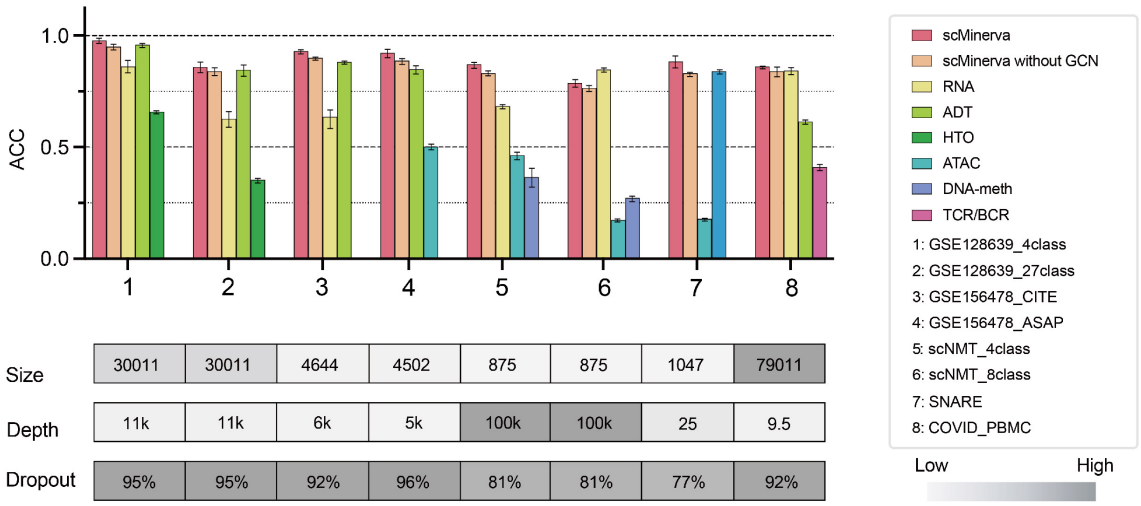

Figure 3: Ablation study results on real-world datasets. The clustered histograms represent results from the same datasets, and below them, we list their sample sizes, measurement depths, and dropout rates. The error bars are the standard deviation of 10 times runs with different random seeds. It shows that scMinerva achieves the best performance in most cases and efficiently integrates valid information from different omics. In some cases, scMinerva without GCN training will produce a lower performance than the best single omics data. However, with a GCN implemented, the performance mostly beats any single omics and is promoted to a higher level.

With good accuracy and anti-noise ability, we further evaluated whether scMinerva could efficiently

capture valid information from different omics and obtains a more comprehensive inference. To achieve this, we perform an ablation study on real-world datasets. The performance of the proposed multi-omics integration method is compared with that of the single-omics data. For each omics, we separately build the graph topology, run the random walk without the omics-transition parameter, generate embeddings, and classify the embeddings with same nearest neighbor classifier fine-tuned by 5% data. To enable a fair comparison, we ran the experiments 10 times for each dataset. Moreover, the data split across all omics for each time is from fixed random seeds with the same proportion and hyper-parameters. Also, to validate the necessity of using GCN, we compared the results before and after training with all other components fixed.

The results in Figure 3 show that with the GCN, combining multi-omics data can obtain better performance than using any one of the single omics data in 8 out of 9 datasets. The only exception is in scNMT for 8-way classification but only has a 3% drop compared to the best-performance omics. It is possibly caused by a serious homogeneity of cell states on the other two omics except for RNA-seq. In this case, the graph topologies from ATAC-seq and DNA-meth are nearly random under a KNN graph construction. Our method might be influenced when start walking from some extremely low-quality nodes. Therefore, in most cases, scMinerva efficiently captures valid information from different omics and obtains a more comprehensive inference.

2.4 scMinerva shows strong scalability on label efficiency

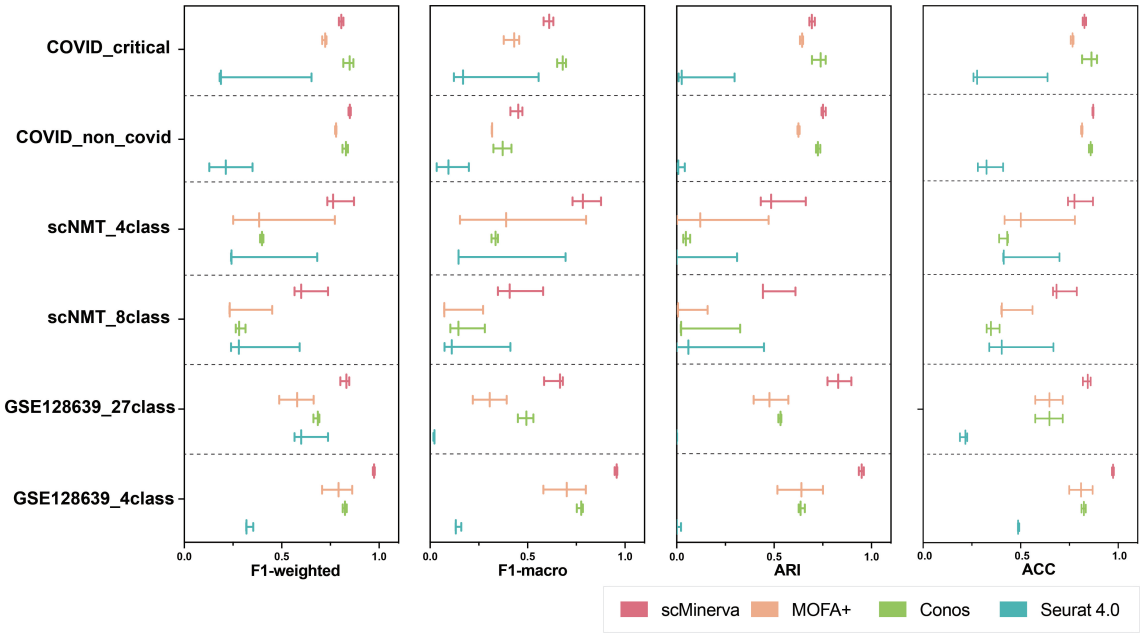

Figure 4: Scalibility evaluation over various label scales of classifier fine-tuning. Each row contains three ticks that represent the integrated classification performances fitted by classifiers fine-tuned by 5%, 10% and 20% data from left to right, respectively. Therefore, the shorter the ticks' length is, the more scalable the method is. Our method achieves overall the best performance and has stronger scalability to the train set size compared to other methods.

Besides the robustness and accuracy, the scalability of methods over different amount used labels when fine-tuning the classifier is also important in practice. Therefore, we fine-tuned classifiers using only 5%, 10%, and 20% annotations as the training set accordingly to fit the output embeddings of all the methods. If a method is scalable and label-efficient, its performance should not drop dramatically while the training size for classifiers are decreased. We evaluated the performance of methods by four metrics: F1-weighted, F1-macro, ARI, and ACC. We evaluated these methods on four triplet-omics datasets.

As shown in Figure 4, we displayed the performance of methods which can handle triplet-omics. Each row contains three ticks that represent a method's performance when fitted with classifiers with 5%, 10% and 20% labels from left to right, respectively. Therefore, if a method is more scalable and more label-efficient, the length of its ticks should be shorter. From Figure 4, it can be easily observed that for all the datasets and train-test splits except the COVID_critical dataset, our method not only

has the best performance but also has relatively small variances compared with others. Especially for the COVID_non_covid dataset and GSE128639 dataset, the performances of scMinerva are not only the best but also the most robust. Its performances are nearly unchanged while the fine-tuning label size is dropped from 20% to 5%. For the COVID_critical dataset, Conos achieves the best performance while our method has comparable performances and smaller various compared to Conos. In summary, scMinerva shows the overall best scalability compared to other methods. The power of scMinerva is sparkled by easy fine-tuning and is not sensitive to the using label size for fine-tuning a separate classifier. This label efficiency reveals great potential in practice for reducing the labor and resource consumption for single-cell data annotation.

2.5 scMinerva Accurately Identify Biomarkers of PBMC Cells

To further prove the versatility of scMinerva, we applied scMinerva on the PBMC multi-omics datasets containing 5 general cell types to analyze the cell type and further identify meaningful biomarkers integratively. Biomarker acts as an indicator of biological processes and plays a vital role in disease detection [33]. With the predictions by scMinerva, we detect biomarkers on the predicted clusters by SCANPY [34]. Experiments are conducted on the GSE128639 dataset, which is a PBMC dataset containing 5 general cell types and 27 subclasses across T cell, B cell, progenitor cell, NK cell, and Mono/DC cell.

We first detected genes that are highly expressed in the 5 general cell types, shown in Figure 5**a**. The X-axis represents the rank of their expression level in this cell type. To further demonstrate the detected genes are more highly expressed in the specific cell types than that in the rest of the cell types, we selected the top 5 marker genes in each cell type and plotted violin graphs for each cell type showing the comparison of the expression level. As shown in Figure 5**b**, the blue color represents the expression of the genes in this cell type, and the orange color represents the sum of expression in the rest cell types. It can be seen that the expression level of the detected marker genes is much higher than that in the rest of the cell types.

From a practical perspective, the detected biomarkers can reveal latent information on their relative biological processes. For example, the gene MALAT1, detected to be ranking 5 in B cells, is shown to be suitable to act as a biomarker, as it correlates with larger tumor size, advanced tumor stage and overall poor prognosis [35]. This evidence mutually confirms the effectiveness of the biomarker detection of scMinerva.

To demonstrate that the prediction of scMinerva can also be used to detect biomarkers in more fine-grained classes, we perform experiments on the 27 subclasses. In Figure 5**c**, the vertical axis represents the 27 subclasses and the horizontal axis indicates the detected marker genes. The color of the violin graphs indicates the expression level of these genes in the corresponding cell types. For example, it can be observed that gene LYZ is highly expressed in cell type GMP, Prog-DC, and cDC2, which is also confirmed by [36, 37]. The detected genes are wildly applied in clinics or research to track the changes in the biological system of cells. For example, the activation of IL7R can initiate precursor B-cell acute lymphoblastic leukemia [38], KLRB1 shows a suppression in human cancer tissues [39], and NKG7 regulates cytotoxic granule exocytosis and inflammation [40].

From the results of biomarker detection, we tried to interpret the model by the following intuition: during the transition and the walk generation, the nodes with a reasonably high feature expression level are like a "Key Opinion Leader" in the graph network. They are normally recognized as anchor nodes in node classification tasks, such as in the KNN algorithm. Therefore, we are curious about their effect during the random walk, especially for the sparse single-cell data which is short of high library-size cells.

To examine this intuition, we compute nodes' occurrence frequency in these walks and check the marker genes' expression level as detected in Figure 5a-c. Since some of the genes are not listed in the raw data, we only contain the reserved genes in the figure. In general, we conclude that, after training, the generated walks as well as the output embeddings, are more obviously influenced by nodes with high feature expression levels. Figure 5**d** is the expression level of the highest occurrence frequency of 20 nodes before training and after training. It is observed that their gene expression levels have no significant differences between the top 20. However, after training, we found that the top 20 nodes are mostly under a high gene expression level.

The comparison strongly reflects that, after training with GCN, nodes with a higher occurrence frequency have a higher probability to upregulate the marker genes compared to the low occurrence frequency nodes. If we assume nodes that have a higher chance to be walked have a higher priority, GCN will assign a higher priority to high expression level nodes and more frequently utilize information from

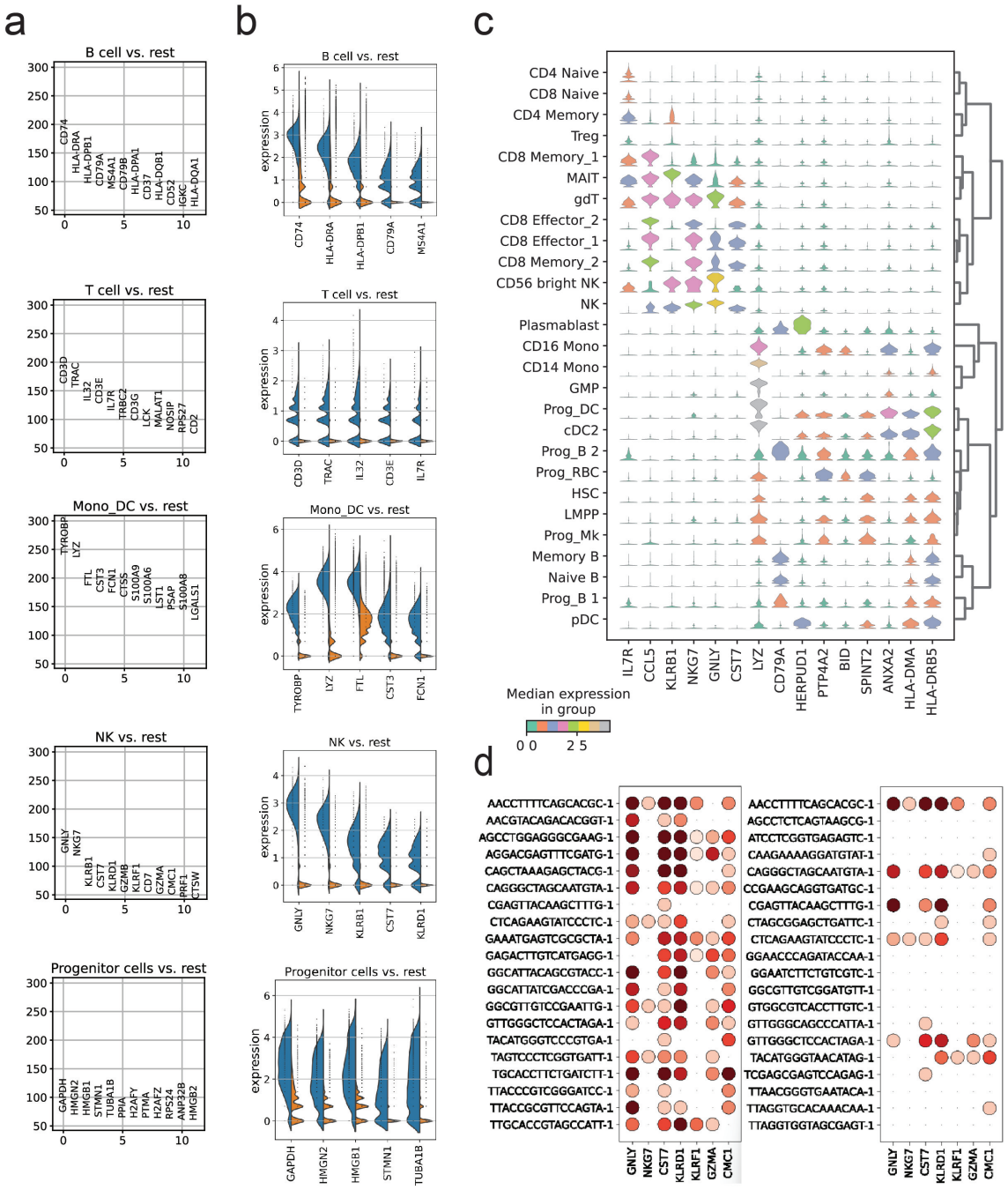

Figure 5: Biomarker detection on PBMC cells from GSE129639. **a**. The detected marker genes in 5 general types. **b**. The comparison of the expression level between the the top 5 detected biomarkers in the corresponding cell type and the rest of the cell types. **c**. Violin graph of the expression level of the biomarkers in fine-grained 27 subclasses. **d**. Nodes expression level on marker gene before training and after training. There is no significant differences between the 20 cells with the most frequent occurrence in walks. But for walks after training. The top 20 nodes show a strong upregulation of marker genes.

its neighbor. These nodes are especially important in the sparse single-cell data as they reserve more valid information concerning their cell type. In another word, our method tends to broaden its knowledge space from these more valuable cells in single-cell data, so that its output is more reasonable as it is strongly benefited by the representatives of different cell types in the graph network. This discovery builds a bridge between our framework to this biological problem.

2.6 scMinerva Reveals Potential Differentiation Changes of Naive Immune Cells after Infection of COVID-19

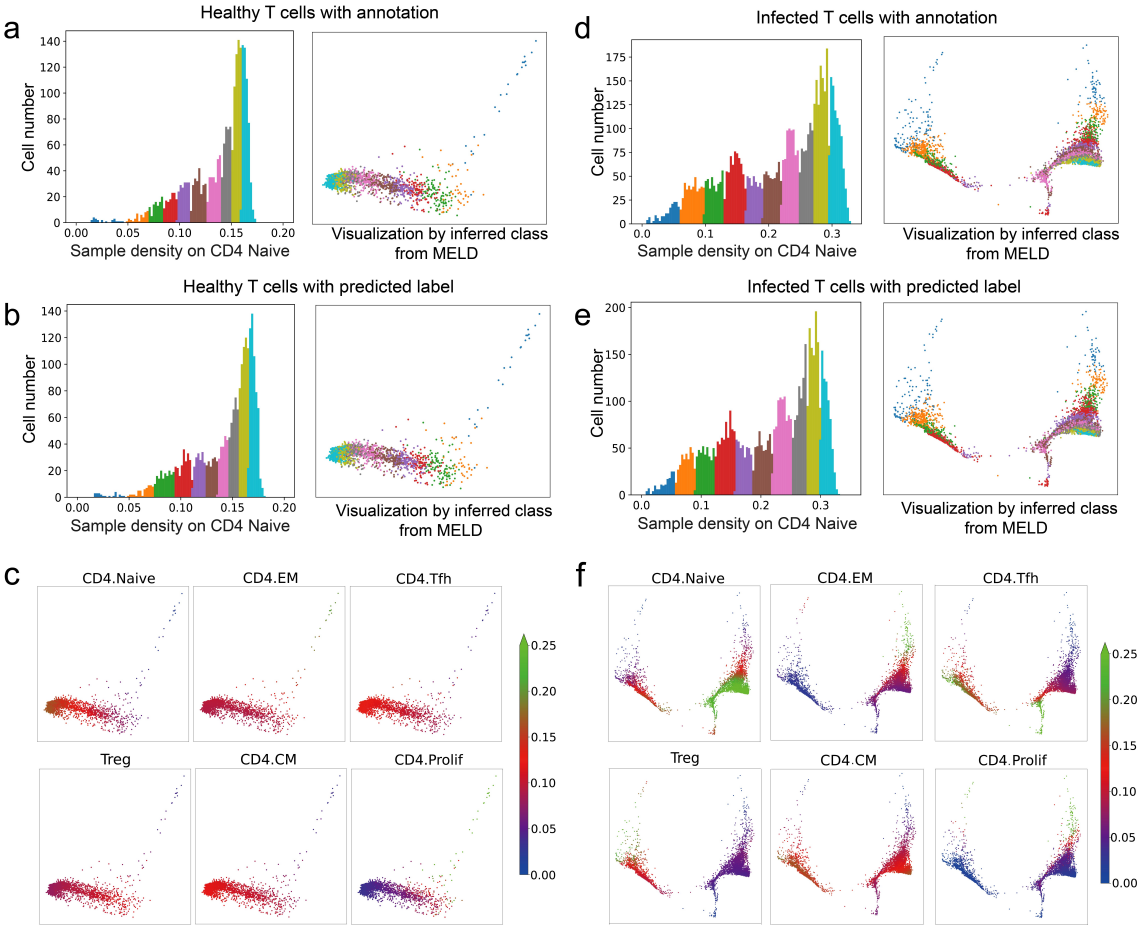

Figure 6: Cell differentiation analysis on CD4 Naive T cells on healthy samples and COVID-19 infected samples. **a**. The clusters fitted by Gaussian Mixture Model(GMM) on healthy tissues inferred by annotations. **b**. Same as **a**, but is inferred by predictions of scMinerva. Our method shows a strong approximation to the results of annotations. **c**. Differentiation likelihood interred by predictions. The differentiation is not active on nearly all cell types. **d**. The clusters fitted by GMM on infected cells inferred by annotations. **e**. Same as **d**, but is inferred by predictions of scMinerva. Our method shows a strong approximation to the annotations. **f**. Differentiation likelihood inferred by predictions. The differentiation to $CD4^+$ T_{CM}, $CD4^+$ T_{FH}, $CD4^+$ prolif are activated.

Since scMinerva has shown its ability to provide biomarkers correctly and the gene activity value on biomarkers can expressively reveal the cell stage in a fine-grained manner, we continued to use the predicted results of scMinerva to analyze the cell differentiation trend at the single-cell level. In this section, we use the predictions from scMinerva to compare the differences in cell differentiation between SARS-CoV-2 (COVID-19) infected and healthy human blood immune cells from the COVID-PBMC dataset [29]. We use the MELD [41] method to analyze the potential differentiation trend of Naive T cells in patients. The selected infected cells are from critical symptoms of human beings and can reflect the changes in the long term. T cells play an important role in human immunity, however, most existing analyses on the impact of COVID-19 on T cells are at a cluster level. Researchers infer the impact from changes in cell-type proportions observed in different symptom durations. In our study, we analyze the potential differentiation tendency of Naive CD4 T cells at a single-cell resolution using MELD and conclude results that confirm some recently proposed hypotheses at a single-cell level. To avoid repetition, we only analyze $CD4^+$ T cells in this section and observe the potential changes in cell differentiation after infection.

Firstly, we analyze healthy cells by inputting the raw expression data and the predicted labels

from scMinerva into MELD to obtain the sample density for each cell in relation to different cell types. The "sample density" is a kernel density estimate that estimates the likelihood of the sample label given the data. The rows represent cells and the columns represent cell types. Each entry in the sample density represents the kernel density estimate for a cell on a specific cell type. In our case, we input 10 unique cell types in the label. Simply, the cell type that a cell obtains the highest sample density is the most likely type that the cell will differentiate to.

Next, we extract only $CD4^+$ Naive cells and run Gaussian Mixture Model (GMM) with the number of components as 10 to sample the density of these cells on $CD4^+$ Naive column. The result is shown in Figure 6**a**. We can observe that most of the cells have a high sample density on $CD4^+$ Naive, and only very few of them are below 0.05. In Figure 6**b**, we repeat the same process but replace the input labels with the predicted labels of scMinerva. It is clear that the inference from our predictions greatly matches the annotations. Following this, in Figure 6**c**, we visualize the sample density score on some important functioning CD4 T cell types with predictions of scMinerva. For healthy cells, there is no active differentiation tendency in most of the cell types. And the sample density from different cell types is all nearly average as shown in Figure 6**c**. We provide a full graph concerning all ten classes in Appendix 9 and 10.

Then, we repeat the procedure on COVID-19-infected cells. Similarly, Figure 6**d,e** compare the GMM modeled sample density on $CD4^+$ Naive column by ground-truth label and predictions of scMinerva respectively. Our prediction also shows a great approximation to the annotations. Following this, we visualized the differentiation tendency on some important functioning CD4 cell types in Figure 6**f**. It can be observed that infected cells are under a prosperous differentiation in comparison to healthy cells, especially to $CD4^+$ prolif, $CD4^+$ T_{FH}, $CD4^+$ T_{EM}, $CD4^+$ T_{CM} and $CD4^+$ T_{reg}. We provide a full graph concerning all ten classes in Appendix.

The result is consistent with the observations from Jung J.H. *et.al.* [42]. They found a significant amount of cells differentiated into diverse memory subsets, comprising $CD4^+$ T_{EM} and $CD4^+$ T_{CM} compared to healthy tissue. Notably, their conclusions are based on the observations at a bulk level. In our study here, we further confirm it from the single-cell level and reveal this phenomenon in the under-differential $CD4^+$ Naive cells. The proliferation is fully supported by other functioning T cells including $CD4^+$ T_H1, $CD4^+$ T_H2 and $CD4^+$ T_{FH} as the full graph shown in Appendix **??**. Cell proliferation is observed in various symptom duration. But it is more obvious for critical patients because of T cell apoptosis [43]. These results reflect the biological system changes and reveal important practical value for benefiting clinical research.

3 Discussion

We developed scMinerva, an unsupervised method for integrated analysis of single-cell multi-omics datasets. The key features distinguishing it from previous methods include 1) A machine-learning based fine-tuning scheme that is less dependent on the results of clustering. For previous unsupervised methods, they finished the cell-type annotation task by clustering the cells first, and then using the aggregated cluster-level expression profiles and the marker genes to label each cluster. Compared to them, scMinerva is less dependent on the clustering performance, which is more powerful for accurate annotation or fine granularity annotation cases. 2) Label-effiicency. Since the embeddings of our method powerfully reserve the global sample (dis)similarities and correlations, they achieve overall excellent performance on single-cell integrated classification even with a classifier fine-tuned with 5% labels. This claimed the label-efficiency of scMinerva. 3) Anti-noise robustness and ability on capturing valid biological information from different omics. Our method performs stably on noisy datasets while other methods are strongly impaired by low-quality omics. scMinerva novelly formulate this integrated analysis task as a heterogeneous graph learning problem and propose a novel algorithm "omics2vec". It works jointly with a GCN model and enables a more comprehensive inference. The analysis of walks generated by omics2vec combined with the knowledge of biomarkers enables an interpretable process to answer why our method could efficiently capture complementary information from different omics. More interestingly, we use the results from scMinerva to identify meaningful biomarkers and analyze single-cell differentiation trends. Our results are consistent with the clinical discoveries and reveal significant potential for biomedicine research.

Although we have shown that scMinerva's integrative classification performance is excellent in many scenarios, we find that our method would perform poorly when handling datasets measured by scNMT technique [31]. But, in the benchmarking process (Figure 2b), the results show that other tools' performance also drops when handling scNMT datasets. One thing interesting is that we evaluated

some fully-supervised methods on this dataset and found that some fully-supervised methods performed well. This phenomenon indicates that this might not be solved well by current unsupervised methods or weakly-supervised methods, and needs to be addressed in future works. In the scenario of clinical data prediction, scMinerva is capable of predicting popular measurement techniques (*i.e.*, CITE, ASAP, *etc.*) in clinical cases stably with statistical power, whose results are consistent with the previous related clinical studies [42, 43].

In summary, scMinerva outperforms previous representative methods and has the ability to flexibly integrate any number of omics types using a simple and independent fine-tuning scheme. We extensively demonstrate scMinerva's excellent performance with respect to data efficiency, anti-noise ability, and robustness, particularly in the face of noisy omics. Additionally, our method is versatile and has been effectively applied to biomarker detection and analysis of single-cell differentiation trends. With scMinerva, researchers can easily identify meaningful biomarkers and analyze single-cell differentiation, among other applications. Considering the fact that it can be integrated with other tools seamlessly, we believe that it will be helpful to investigate the connection between the single-cell data and the numerous clinical discoveries.

4 Method

4.1 Datasets, Annotations adn Preprocessing

In this section, we provide detailed information on the data preprocessing and the cell-type information for each dataset used in our analysis.

Table 2: Datasets analyzed in this paper.

Dataset	Cell Number	Cell Type	#Omics	Depth	Dropout
GSE128639	30k	PBMC	3	11k	95%
GSE156478-CITE	4.6k	PBMC	2	6k	92%
GSE156478-ASAP	4.5k	PBMC	2	5k	96%
COVID-PBMC	79k	PBMC	3	100k	81%
scNMT	875	Mouse Embryonic Stem Cells	3	25k	77%
SNARE	1047	Mouse Cerebral Cortices	2	9.5k	92%

GSE128639 The multi-omics data matrices were used as quantified in the original experiments [28]. For gene expression, standard log-normalization with default parameters in Seurat [28] was conducted. The only difference with the original implementation in the paper is that we take the raw data of HTO separately from the dataset as the third omics. HTO is extremely sparse data so with this as a third omics, the performance of Seurat 4.0 will be strongly lagged back. The cell-type information took into consideration of both known RNA and protein markers. They placed clusters into eight broad groups and further subdivided these groups into 30 level 2 annotation categories. For the details, please refer to the appendix of the reference [28].

GSE156478 For the GSE156478 dataset, we followed the same data and cell-type information processing as presented in scJoint [16]. To refer to the detail, please check its data preprocessing section in the appendix. Briefly, the control and stimulated CITE-seq were filtered based on the following criteria: mitochondrial reads greater than 10%; the number of expressed genes less than 500; the total number of UMI less than 1000; the total number of ADTs from the rat isotype control greater than 55 and 65 in the control and stimulated conditions respectively; the total number of UMI greater than 12,000 and 20,000 for the control and stimulated conditions respectively; the total number of ADTs less than 10,000 and 30,000 for control and stimulated conditions respectively. The cells that were classified as doublets in the original study were filtered out. For the ASAP-seq data, cells with a number ADTs more than 10,000 and number of peaks more than 100,000 were filtered out. Finally, 4502 cells (control) and 5468 cells (stimulated) from ASAP-seq, 4644 cells (control), and 3474 cells (stimulated) from CITE-seq were included in the downstream analysis. The number of common genes across the four matrices is 17441 and the number of common ADTs is 227 [16].

scNMT The multi-omics data and cell-type information were obtained from the original study [44]. Generally speaking, gene counts were quantified from the mapped reads by featureCounts [45], and gene annotations were pbtained from Ensembl version 87 [46]. Only protein-coding genes mathcing canonical chromosomes were considered. For methylation and accessibility pseudo-bulk profiles, the values were averaged using running windows of 50 bp. The information from multiple cells was combined by calculating the mean and the standard deviation for each running window. Accessibility profiles were processed with each cell and gene in +/- 200 bp windows around the TSS. Only genes covered in at least 40% of the cells with a minimum coverage of 10 GpC sites were considered [31].

SNARE SNAREseq [30] consists of chromatin accessibility and gene expression. The data is collected from a mixture of human cell lines: BJ, H1, K562, and GM12878. We reduce the dimension of the data by PCA. The size of the resulting matrix for scATAC-seq is of 1047×1000 and 1047×500 for the gene matrix. We use the code provided by the author to generate annotations for BJ, H1, K562, and GM12878. The cell-type information was obtained from the original study [30].

COVID-PBMC The data and cell-type information were obtained from the original study [47]. Briefly, FASTQ files were generated from raw sequencing reads by the Cell Ranger mkfastq pipeline. Cell Ranger count pipeline (v3.1) was utilized to perform alignment, filtering. barcode counting, and UNI counting. GRCh38 was denoted as genome reference. To remove dead and dying cells, Cells with mitochondrial gene percentages higher than 12% and cells with less than 200 genes was filtered out. For CITE-seq samples, the cells were demultiplexed and hashing adt COUNTS were removed. The remaining counts were normalized by library size and square. For TCR data, the raw sequencing reads of the T cell receptor (TCR) libraries were prcessed by the Cell Ranger V(D)J pipeline by 10x Genomics. Only V(D)J contigs with high confidence defined by cell ranger were considered. The cells of one beta chain contig and zero or one alpha chain contig were remained [47].

4.2 The scMinerva Framework

4.2.1 Data Simulation from Real-world Datasets

We generate a four-omics dataset based on the synthetic RNA-seq generated by Splatter [26]. To maintain the mapping between different modalities, we train three Feedforward Neural Networks (FNN) to simulate the mapping between different modalities, including mappings from scATAC-seq to scRNA-seq, from scRNA-seq to ADT matriX, and from scATAC-seq to ADT matrix. We utilize the real-world datasets mentioned below to train these three GNNs. Firstly, we create synthetic scRNA-seq by Splatter, the dimension of which is equal to scRNA-seq from sci-CAR [3]. Then we map the generated scRNA-seq to scATAC-seq by the FNN we trained upon sci-CAR. Similarly, we generate two other ADT matrices from simulated scRNA-seq and scATAC-seq. These two models are trained with scRNA-seq, scATAC-seq, and ADT matrices from GSE156478 and we utilize PCA to make the dimension of scRNA-seq and scATAC-seq consistent with sci-CAR so that we can generate a set of data. We developed four sets of data, with five classes and sample numbers 2k, 5k, 10k, and 30k, respectively. The simulated RNA, ATAC, ADT from RNA, and ADT from ATAC data are of feature numbers 815, 2613, 227, and 227 respectively.

4.2.2 Preliminaries

Consider a dataset with c omics and n samples (*i.e.*, cells). Denote the samples set $S = \{s_i\}, i \in [1, n]$. We consider the sample cells as nodes, the gene activity value of sample cells in the omics as node features and the similarities (*i.e.*, Euclidean distance) between nodes pairwisely as their edge weights. The biological problem is formulated in such a graph setting. Throughout the paper, $\mathcal{G}$ stands for a graph, and $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ refers to its node set and $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V}, \mathbb{R})$ is its edge list. $\mathbf{X} \in \mathbb{R}^{|\mathcal{E}| \times F}$ stands for the feature matrix where F is the amount of features and a node embedding $\mathbf{x}_i$ is a row of $\mathbf{X}$ of dimension $\mathbb{R}^F$. The adjacency matrix of $\mathcal{G}$ is defined as $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ where $\mathbf{w}_{v_a v_b}$ stands for the weight of the connecting edge from node v_a to node v_b.

Our objective is to learn a unified embedding for each sample in S integrating all the omics. To learn the embedding, We will first construct a heterogeneous graph $\mathcal{G}$ from a list of sub-graphs $\{\mathcal{G}^{(i)}\}$. Following the above scheme, we define the corresponding subgraph for the j-th omics as $\mathcal{G}^{(j)}$ and define $\mathcal{V}^{(j)}, \mathcal{E}^{(j)}, \mathbf{X}^{(j)}, \mathbf{W}^{(j)}$ accordingly. For the heterogeneous graph, we define it as $\mathcal{G}$ and define $\mathcal{V}, \mathcal{E}, \mathbf{X}, \mathbf{W}$ accordingly.

Algorithm 1 Pseudo-code of scMinerva

Input: node sets $\{\mathcal{V}^{(j)}\}$, feature matrices $\{\mathbf{X}^{(j)}\}$, transition parameters p, q and z, GCN networks $f_{GCN}(\cdot,\cdot)$.
$\mathbf{W}^{(j)} \leftarrow$ Build adjacency matrix by computing the Euclidean distance of node features in $\mathbf{X}^{(j)}$ pairwisely
$\mathcal{E}^{(j)} \leftarrow$ Build edge lists for subgraphs by weighted K-nearest neighbor algorithm
$\mathcal{G} \leftarrow$ Construct heterogeneous graph from $\{\mathcal{G}^{(j)}\}, \{\mathbf{W}^{(j)}\}$
$\mathcal{G} \leftarrow$ Normalize the transition probability with p, q, z as Eq. (1) and update the alias edge weights
Walks $\mathbf{J} \leftarrow$ Run random walk on $\mathcal{G}$ ▷ Omics2vec for Nodes
$\mathbf{X} \leftarrow$ Input $\mathbf{J}$ into the word2vec model and optimize Eq. (4) to obtain features of node embeddings
while the stopping criteria is not satisfied **do**
    $\mathbf{X} \leftarrow f_{GCN}(\mathbf{X}, \mathbf{W})$
    if iterate t mod $b = 0$ **then** ▷ $b \leftarrow$ Loss update interval
        Compute the centroid matrix $\boldsymbol{\nu}$ and cluster nodes by k-means
        $P \leftarrow \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_{j'} (q_{ij'}^2 /) \sum_i q_{ij'}}$
    end if
    $Q \leftarrow \frac{(1+||\mathbf{x}_i - \boldsymbol{\nu}_i||^2)^{-1}}{\sum_{j'} (1+||\mathbf{x}_i - \boldsymbol{\nu}_{j'}||^2)^{-1})}$
    LOSS $\leftarrow$ KL$(P||Q)$ ▷ Unsupervised Loss
    Update network f_{GCN} to minimize LOSS
end while
$\mathbf{W} \leftarrow$ Build adjacency matrix by computing the Euclidean distance of node features in $\mathbf{X}$ pairwisely
$\mathcal{G} \leftarrow$ Normalize the transition probability with p, q, z as Eq. (1)
$\mathbf{J}_{samples} \leftarrow$ Random walk on $\mathcal{G}$ and map the nodes of $\mathbf{J}$ from the same sample cells to a same index
$\mathbf{X}_{samples} \leftarrow$ Input $\mathbf{J}_{sample}$ into the word2vec model and obtain features of sample embeddings
return Sample embedding $\mathbf{X}_{samples}$

4.2.3 Heterogeneous Graph Construction

The problem is formulated as a graph learning problem and to begin, a heterogeneous graph $\mathcal{G}$ is initialized as the input. Given the feature matrices $\{\mathbf{X}^{(j)}\}$ and node sets $\{\mathcal{V}^{(j)}\}$ for each omics, the Euclidean distances between the nodes are calculated pairwisely to form $\{\mathbf{W}^{(j)} \mid \mathbf{W}^{(j)} \in \mathbb{R}^{n \times n}\}$. The K-nearest neighbor algorithm is then used to construct the edge lists $\{\mathcal{E}^{(j)}\}$ and define the list of subgraphs $\{\mathcal{G}^{(j)}\}$.

To aid further discussion, the term $\mathbf{o}(v)$ is defined as the *omics index function* of node v and v_{rj} represents the mapping node of sample v_r in the j-th omics, *i.e.*, $\mathbf{o}(v_{rj}) = j$.

The heterogeneous graph $\mathcal{G}$ is then constructed, containing all nodes and edges from $\{\mathcal{V}^{(j)}\}, \{\mathcal{E}^{(j)}\}$. To facilitate inter-omics transitions, initial values of the omics-to-omics edges are assigned to be the unity. Formally speaking, the adjacency matrices $\{\mathbf{W}^{(j)}\}$ are appended diagonally to form $\mathbf{W} \in \mathbb{R}^{nc \times nc}$ and nodes from the same sample in different omics are linked with an edge weight of 1. Denote $\mathbf{w}_{ab}$ as the edge weight between node a and node b and its value is assigned by the (a-th, b-th) entry of matrix $\mathbf{W}$. Formally, $\mathbf{w}_{v_{rt}v_{rl}} = 1$ for all v_{rt} and v_{rl} where $t \neq l$. Undefined entries in $\mathbf{W}$ are assigned a small enough constant, $\epsilon = 1e-4$, to prevent zero division. The new graph has $|\mathcal{V}| = \sum_{j=1}^{c} |\mathcal{V}^{(j)}|$ nodes and $|\mathcal{E}| = \sum_{i=1}^{c} |\mathcal{E}^{(i)}| + 2 \cdot \frac{nc(c-1)}{2}$ edges, with $\mathcal{G}$ being a directed graph, particularly for the omics-to-omics edges (note the coefficient of 2). In the next steps, the adjacency matrix $\mathbf{W}$ and the feature matrix $\mathbf{X}$ of $\mathcal{G}$ will be optimized.

4.2.4 Omics2vec

Now we will introduce our *omics2vec* algorithm, which is inspired by word2vec [48] and node2vec [22]. It fits node2vec to this biological problem and enables integrated analysis on the heterogeneous graph built on multi-omics data. To briefly introduce the background, word2vec is a model that learns to embed words in a high-dimensional space where semantically similar words are close to each other [48]. Furthermore, node2vec extends the idea of word2vec to graph-structured data [22]. It learns to embed nodes by defining a biased random walk procedure that explores the graph nodes' similarity in a way that balances the exploration of local and global network structures. By viewing different nodes in walks as words, node2vec inputs the generated walks into word2vec and ensures the embeddings of spatially close nodes are similar to each other.

Node2vec is a graph search algorithm that combines Breadth-First Search (BFS) and Depth-First-Search (DFS) techniques. It introduces two hyper-parameters, p and q, to control the second-order random walk. The parameter p controls the probability of revisiting a node, while q controls the in-out forward (for the detailed geometric meaning of these parameters, please refer to Eqt. (2)). We further investigate the parameter sensitivity of this algorithm, please refer to Section 10.1. By multiplying the multiplication inverses of these hyper-parameters with the edge weights and normalizing the transition probability to 1, node2vec sets up a biased random walk procedure. However, applying node2vec to this biological heterogeneous graph presents a challenge because the inter-omics transition edges are undefined. Moreover, the weights of the inter-omics edges play a significant role in the performance, as they determine the probability of being explored in the neighborhoods of a sample's counterparts on different omics. To overcome this challenge, we propose a novel algorithm called omics2vec, which extends the node2vec idea to analyze biological heterogeneous graphs. Omics2vec enables inter-omics transition and can work together with the GCN model training (Section 4.2.5), which optimizes inter-omics edge weights. To compute the transition probability on graph $\mathcal{G}$, we combine the edge weights $\mathbf{W}$ and transition-control hyper-parameters, generate random walks based on these probabilities, and input them into the word2vec model to generate node embeddings. Omics2vec has two types: embed for nodes and embed for the samples (single cells). The two types share the same first few steps as the following illustration:

To represent the transition probability from one node b to another node a, a transition probability function $P(a|b)$ is defined. To generate random walks of a fixed length l in graph $\mathcal{G}$, we denote α_i as the i-th step in the walk α starting from α_0. Steps $\alpha_g \mid g \in [1, l]$ are generated by the following probability function:

$$P(\alpha_g = v_a | \alpha_{g-1} = v_b) = \begin{cases} \frac{\pi_{v_a v_b}}{Z} & \text{if} (v_a, v_b) \in \mathcal{E}, \\ 0 & \text{otherwise}, \end{cases} \tag{1}$$

where $\pi_{v_a v_b}$ is the unnormalized transition probability from node v_b to node v_a that is defined in the following contents and Z is a normalizing constant.

The normalizing constant Z is carefully selected to guide the exploration of different types of neighborhoods. Instead of the mixture of BFS and DFS used by node2vec, we guide the biased random walk with three parameters p, q, and z all of whom are positive values. The parameter z is a new addition to the algorithm and can be seen as the "omics-first search".

Considering a random walk that just traveled from node v_1 (*i.e.*, step $\alpha_{g-1} = v_1$) to node v_2 (*i.e.*, step $\alpha_g = v_2$), the walk now is determining the next step α_{g+1}. So it evaluates the transition probability $\pi_{v_2 v_3}$ on edge (v_2, v_3) leading from v_2. We set the unnormalized transition probability as $\pi_{v_2 v_3} = \alpha_{pqz} \cdot \mathbf{w}_{v_2 v_3}$. Here, α_{pqz} is a function that depends on the hyperparameters p, q, and z, and *the shortest distance* $d_{v_1 v_3}$ between nodes v_1 and v_3 in the graph:

$$\alpha_{pqz}(v_2, v_3) = \begin{cases} \frac{1}{p} & \text{if } \mathbf{o}(v_1) = \mathbf{o}(v_3) \text{ and } d_{v_1 v_3} = 0, \\ 1 & \text{if } \mathbf{o}(v_1) = \mathbf{o}(v_3) \text{ and } d_{v_1 v_3} = 1, \\ \frac{1}{q} & \text{if } \mathbf{o}(v_1) = \mathbf{o}(v_3) \text{ and } d_{v_1 v_3} = 2, \\ \frac{1}{z} & \text{if } \mathbf{o}(v_1) \neq \mathbf{o}(v_3). \end{cases} \tag{2}$$

Specifically, if nodes v_1 and v_3 belong to the same omics type and have a shorter distance of 0, then $\alpha_{pqz} = \frac{1}{p}$. If p is smaller, it means that the algorithm has a higher probability of revisiting the node in the last step. If the distance between v_1 and v_3 is 1, then $\alpha_{pqz} = 1$, indicating that the algorithm can explore nodes in the same omics type that are immediate neighbors. If the distance between v_1 and v_3 is 2, then $\alpha_{pqz} = \frac{1}{q}$. This setting allows the algorithm to explore nodes in the same omics type that are not immediate neighbors. Finally, if v_1 and v_3 belong to different omics types, then $\alpha_{pqz} = \frac{1}{z}$. This setting allows the algorithm to explore nodes and moreover, their neighborhood, in other omics types. Once the transition probabilities are calculated, the algorithm generates walks randomly, starting from each source node and repeating the random walk r times for each source node. The number r is a hyperparameter that determines the number of walks generated from each source node.

For the next few steps, omics2vec for nodes and omics2vec for samples are different. Omics2vec for nodes will directly input the generated walks into the word2vec model which outputs the node

embeddings. However, omcis2vec for samples will conduct an additional index mapping step for generated walks which maps the node indices to their sample indices. After the mapping, we obtain the walks that reflect the sample's similarities in the perspective of the global heterogeneous graph. By inputting the walks for samples after mapping, we obtain the embeddings for sample cells. Next, we will elaborate on these steps in more detail.

Omics2vec for Nodes When the random walk finishes, these are $|\mathcal{V}|\cdot r$ walks in total. Since each walk is of length l, the walk matrix $\mathbf{J} \subseteq \mathbb{N}^{(|\mathcal{V}|\cdot r)\times l}$ is the input of the word2vec model. Let $f : S \to \mathbb{R}^F$ be the mapping function from the node set to embeddings of dimension F. We aim to learn such representations for the later tasks. For source node, $v_s \in \mathcal{V}$, define $N(v_s) \subset \mathcal{V}$ are the neighborhood of node v_s generated through a sample of walks. With a skip-graph architecture, we are trying to find the f that gives

$$\max \sum_{v_s \in \mathcal{V}} \log Pr(N(v_s)|f(v_s)). \tag{3}$$

The above objective function maximizes the log-probability of observing $N(v_s)$ for node v_s conditioned on its mapping after function f. Here, by assuming conditional independence among observing different neighborhood nodes given the feature representation of the source, Eq. (3) can be simplified to:

$$\max_f \sum_{v_s \in \mathcal{V}} [-\log K_{v_s} + \sum_{v_{n_i} \in N(v_s)} f(v_{n_i}) \cdot f(v_s)]. \tag{4}$$

where $K_{v_s} = \sum_{v \in \mathcal{V}} \exp(f(v_s) \cdot f(v_s))$ is the partition function for nodes. Solve Eq. (4) using stochastic gradient ascent over the model defining the features f. The output embeddings from f is the feature matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times F}$ of nodes on the heterogeneous graph $\mathcal{G}$. After obtaining the feature matrix $\mathbf{X}$, we can improve it using graph convolutional networks (GCNs). GCNs are a type of neural network that operates on graph-structured data and can learn representations by aggregating information from the node's local neighborhood. By applying GCNs to the feature matrix $\mathbf{X}$, we can learn more expressive node representations that take into account the structure of the heterogeneous graph.

Omics2vec for Samples It is similar to the Omics2vec for Nodes algorithm. However, in the final step of our method, we use the walks generated by omics2vec to map the indices of nodes from different omics to their indices of sample cells. Specifically, we ensure that the counterparts of one sample cell from different omics are mapped to the same sample cell in the walks. This remapping produces a new set of walks, denoted as $\mathbf{J}_{sample}$, which contain information about the relationships between sample cells across multiple omics. Next, we input the remapped walks $\mathbf{J}_{sample}$ into the word2vec model to obtain the embedding of sample cells $\mathbf{X}_{sample}$. This embedding captures the integrated multi-omics information by leveraging the well-clustered neighborhood information from the entire heterogeneous graph.

4.2.5 Model Training

With the node features $\mathbf{X}$ as well as the graph adjacency matrix $\mathbf{W}$ as inputs, we train a two-layer GCN model $f_{GCN}(\cdot,\cdot)$ to optimize the graph. To train the GCN, we adopt an unsupervised clustering loss function DeepCluster [49] which produces pseudo-labels and learns from the iteratively updated cluster assignments. The training enforces the nodes with higher similarity to be closer in the embedding space and the nodes with low similarity to be dispersed.

To apply this loss, we first run k-means algorithms to cluster the output feature matrix $\mathbf{X}$ into k different groups based on the geometric neighborhood. K-means outputs a set of optimal pseudo-labels and we denote it as $\{y^*\}$. Our loss supervising GCN is based on minimizing the Kullback-Leibler (KL) divergence between a student's t-distribution kernel Q to the clusters and a target distribution P. The target distribution P is a Gaussian distribution centered at each cluster center. The student's t-distribution kernel Q is used to calculate the soft assignment probability q_{ij} of the embedding $\mathbf{x}_i$ to the cluster centroid $\boldsymbol{\nu}_i$, $q_{ij} = \frac{(1+||\mathbf{x}_i-\boldsymbol{\nu}_i||^2)^{-1}}{\sum_{j'}(1+||\mathbf{x}_i-\boldsymbol{\nu}_{j'}||^2)^{-1})}$. Next, based on q_{ij}, a target distribution P is calculated to help learn from the assignments with higher scores where $p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_{j'}(q_{ij'}^2/)\sum_i q_{ij'}}$. Finally, the loss function is defined as

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{5}$$

We optimize the feature matrix $\mathbf{X}$ by minimizing Equation 5. We continue to iterate until the loss is small enough (*i.e.*, 5e-2), at which point we terminate the iteration and output the embeddings $\mathbf{X}$ for nodes in $\mathcal{G}$. By optimizing the feature matrix, we ensure that similar nodes from the heterogeneous graph are gathered together in the embedding space. Once we have the node embeddings, we compute the distance of nodes pairwise and reconstruct the heterogeneous graph from these embeddings. The omics-transition edges are updated by the results after GCN training, resulting in more reasonable edges than the naive initialization (assigning a unity value to all the inter-omics edges). These updated edges can improve downstream performance as shown in the ablation study (Section **??**). Then, we run Omics2vec for Samples to generate walks following the process in Section 4.2.4 and obtain the sample embeddings. Then we will evaluate the model through multiple experiments.

4.2.6 Experiment Settings

Cell Classification Previous methods of cell classification tasks mostly are conducted in a fully-supervised manner. As we have mentioned in the introduction, they could not quickly be adapted to fit different grain-level annotations. Jointly considering the labor-intensive and time-consuming processing of data annotation, we evaluate the label-efficiency of methods with a simple nearest-neighbor classifier which is independent of the unsupervised training stage. In detail, for the produced embeddings of scMinerva and other methods, we fit them with independent nearest neighbor-based classifiers and fine-tune them using only 10% annotations of the whole training set.

We implement existing state-of-the-art methods with similar approaches as illustrated in our introduction, including DeepMAPS, CiteFuse, totalVI, and Seurat 4.0 (weighted nearest neighbor), in their recommended settings. For MOFA+ and Conos, the dimension of their embeddings needs to be enlarged to fit a large dataset. So we search the embedding dimension of $\{100, 200, 300, 400\}$ and select the best-performing dimension. Notably, MOFA+ is very memory-consuming, and we failed to conduct the experiment with the originally preprocessed data using a RAM of 32GB when the number of samples is greater than 10k. Thus, we perform PCA to reduce the input dimension, where we also search among $\{100, 200, 400, 800\}$ to select the optimal performance. Besides, DeepMAPS, CiteFuse, and TotalVI cannot process three-omics data, and therefore they are excluded in the chart for COVID-PBMC, scNMT, and GSE128639. To evaluate the quality of the generated embeddings, we perform classification by fitting a K-nearest Neighbor (KNN) Classifier with the number of neighbors as 30 for datasets containing more than 5k samples, and with the number of neighbors as 8 for datasets smaller than 5k.

Cell Differentiation Analysis Cell differentiation is the process by which a single cell develops into many different specialized cell types with unique functions. It is important for understanding cell development, disease, and drug and for developing new therapies and treatments. In this paper, we take cell differentiation analysis as an example to show the practice value of our model together with MELD [36]. We take human blood immune cells from dataset COVID-PBMC [50] to compare the differences in cell differentiation between cells infected with SARS-CoV-2 (COVID-19) and healthy cells. To avoid repeating, we only take $CD4^+$ T cells which contain 10 sub-classes to observe the potential cell differentiation changes. Since there are 10 sub-classes, we run the Gaussian Mixture Model (GMM) with the number of components as 10 and all other parameters are the same as the default setting in MELD.

Model Interpretation Loading.

4.3 Hyperparameters

Random Walk We run random walk on the heterogeneous graph with three transition controlling parameters named p, q, and z, where p controls the likelihood of immediately revisiting a node in the walk, q allows the search to differentiate between "inward" and "outward" nodes, and z controls an inter-omics transition within the frame. By default, we set p, q, z all equal to the unity. To ensure a rich connection between omics and avoid zero division during normalizing, we also introduce a hyper-parameter δ to smooth the graph. In another word, the omics-transition links will have a small enough default value equal to δ before normalizing. We set δ to $1e-4$.

For other random walk parameters, we follow the default value set in node2vec [22]. In detail, on each node of the graph, we run random walk start from it 10 times. Each time it will generate a walk of

length 80. With the generated walks, we input them to word2vec under algorithm CBOW and window size 10. Word2vec will output embeddings of dimension 128 for nodes contained in walks.

GCN Model The GCN architecture has two primary hyperparameters: the latent dimensions of the autoencoders and the k of k-means that DEC clusters the nodes. We simply set $k = c \times \#\text{cell-type}$ (number of cell-types) and stop the training while the loss function achieves a value smaller than $5e-2$. The dimension of the latent space is set to be 32.

4.4 Performance Evaluation

Accuracy (ACC), F1-score, F1-macro, and adjusted rand index (ARI) are used to evaluate the classification performance of models. Each of the three measures has a different focus. ACC measures the precision of the results. F1-score seeks a balance between precision and recall. ARI emphasizes more on the cluster level which is a measure of the similarity between two data clusters. F1 score can be calculated [51] as F1 score= $\frac{TP}{TP+\frac{1}{2}(FP+FN)}$; the F1-macro is the arithmetic mean of all the per-class F1 scores; ARI is used to measure the similarity between the predicted labels and ground truth. It counts pairs that are assigned in the same or different clusters in the predicted and actual clusters. ARI is a corrected-for-chance version of the Rand index defined as $\text{ARI} = \frac{\text{RI}-\mathbb{E}[\text{RI}]}{\max[\text{RI}]-\mathbb{E}[\text{RI}]}$ where $\text{RI} = \frac{\text{TP+TN}}{\text{TP+TN++FP++FN}}$ and $\mathbb{E}$ is the expectation value.

4.5 Statistics and Reproducibility

Datasets were chosen in order to show the functionality and performance of our method. No data were excluded from the analyses. Replication and randomization are not applicable since we did not collect any experimental data. Hypothesis testing methods are explained in each figure legend. To reproduce the results, please find the Source Data file we provided.

4.6 Software Comparison and Settings

Loading

5 Funding

This research is funded by the Chinese University of Hong Kong (CUHK) with the award number 4937025, 4937026, 5501517, and 5501329.

Yongshuo Zong was supported by the United Kingdom Research and Innovation (grant EP/S02431X/1), UKRI Centre for Doctoral Training in Biomedical AI at the University of Edinburgh, School of Informatics. For the purpose of open access, the author has applied a creative commons attribution (CC BY) licence to any author-accepted manuscript version arising.

6 Data Availability

The datasets analyzed in this study are available under the following accession numbers: GSE128639 [28], GSE156478-CITE [27], GSE156478-ASAP [27]), COVID-PBMC [29]), SNARE-seq [30] and scNMT-seq [31].

7 Code Avalibility

The open-source implementation of scMinerva is available at https://github.com/yistyu/scMinerva, and the experiments conducted to produce the main results of this article are also stored in this repository.

8 Ethics approval and consent to participate

Not applicable.

9 Competing interests

The authors declare that they have no competing interests.

10 Authors' contributions

T.Y. and Y.L. conceived the project. T.Y. implemented the model. T.Y. and Y.W. generated figures. T.Y., Y.Z., Y.W., and X.W. run the experiments on baseline methods. T.Y. and Y.Z. wrote the manuscript with feedback from all authors. Y.L. supervised the project. The authors read and approved the final manuscript.

References

[1] Stoeckius, M., Hafemeister, C., Stephenson, W., Houck-Loomis, B., Chattopadhyay, P.K., Swerdlow, H., Satija, R., Smibert, P.: Simultaneous epitope and transcriptome measurement in single cells. Nature methods **14**(9), 865–868 (2017)

[2] Pott, S.: Simultaneous measurement of chromatin accessibility, dna methylation, and nucleosome phasing in single cells. Elife **6**, 23203 (2017)

[3] Cao, J., Cusanovich, D.A., Ramani, V., Aghamirzaie, D., Pliner, H.A., Hill, A.J., Daza, R.M., McFaline-Figueroa, J.L., Packer, J.S., Christiansen, L., *et al.*: Joint profiling of chromatin accessibility and gene expression in thousands of single cells. Science **361**(6409), 1380–1385 (2018)

[4] Zhu, C., Preissl, S., Ren, B.: Single-cell multimodal omics: the power of many. Nature methods **17**(1), 11–14 (2020)

[5] Lähnemann, D., Köster, J., Szczurek, E., McCarthy, D.J., Hicks, S.C., Robinson, M.D., Vallejos, C.A., Campbell, K.R., Beerenwinkel, N., Mahfouz, A., *et al.*: Eleven grand challenges in single-cell data science. Genome biology **21**(1), 1–35 (2020)

[6] Hicks, S.C., Townes, F.W., Teng, M., Irizarry, R.A.: Missing data and technical variability in single-cell rna-sequencing experiments. Biostatistics **19**(4), 562–578 (2018)

[7] Argelaguet, R., Velten, B., Arnol, D., Dietrich, S., Zenz, T., Marioni, J.C., Buettner, F., Huber, W., Stegle, O.: Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. Molecular systems biology **14**(6), 8124 (2018)

[8] Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J.C., Stegle, O.: Mofa+: a statistical framework for comprehensive integration of multi-modal single-cell data. Genome biology **21**(1), 1–17 (2020)

[9] Jin, S., Zhang, L., Nie, Q.: scai: an unsupervised approach for the integrative analysis of parallel single-cell transcriptomic and epigenomic profiles. Genome biology **21**(1), 1–19 (2020)

[10] Gayoso, A., Steier, Z., Lopez, R., Regier, J., Nazor, K.L., Streets, A., Yosef, N.: Joint probabilistic modeling of single-cell multi-omic data with totalvi. Nature methods **18**(3), 272–282 (2021)

[11] Stuart, T., Srivastava, A., Lareau, C., Satija, R.: Multimodal single-cell chromatin analysis with signac. BioRxiv (2020)

[12] Kim, H.J., Lin, Y., Geddes, T.A., Yang, J.Y.H., Yang, P.: Citefuse enables multi-modal analysis of cite-seq data. Bioinformatics **36**(14), 4137–4143 (2020)

[13] Barkas, N., Petukhov, V., Nikolaeva, D., Lozinsky, Y., Demharter, S., Khodosevich, K., Kharchenko, P.V.: Joint analysis of heterogeneous single-cell rna-seq dataset collections. Nature methods **16**(8), 695–698 (2019)

[14] Duren, Z., Chang, F., Naqing, F., Xin, J., Liu, Q., Wong, W.H.: Regulatory analysis of single cell multiome gene expression and chromatin accessibility data with screg. Genome biology **23**(1), 1–19 (2022)

[15] Ma, A., Wang, X., Wang, C., Li, J., Xiao, T., Wang, J., Li, Y., Liu, Y., Chang, Y., Wang, D., et al.: Deepmaps: Single-cell biological network inference using heterogeneous graph transformer. bioRxiv (2021)

[16] Lin, Y., Wu, T.-Y., Wan, S., Yang, J.Y., Wong, W.H., Wang, Y.: scjoint integrates atlas-scale single-cell rna-seq and atac-seq data with transfer learning. Nature Biotechnology, 1–8 (2022)

[17] Shi, W., Zhao, X., Chen, F., Yu, Q.: Multifaceted uncertainty estimation for label-efficient deep learning. Advances in neural information processing systems **33**, 17247–17257 (2020)

[18] Huang, S.-C., Shen, L., Lungren, M.P., Yeung, S.: Gloria: A multimodal global-local representation learning framework for label-efficient medical image recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3942–3951 (2021)

[19] Zhao, Z., Yang, G.: Unsupervised contrastive learning of radiomics and deep features for label-efficient tumor classification. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part II 24, pp. 252–261 (2021). Springer

[20] Wei, Z., Zhang, S.: Callr: a semi-supervised cell-type annotation method for single-cell rna sequencing data. Bioinformatics **37**(Supplement_1), 51–58 (2021)

[21] Seal, D.B., Das, V., De, R.K.: Cassl: A cell-type annotation method for single cell transcriptomics data using semi-supervised learning. Applied Intelligence, 1–19 (2022)

[22] Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864 (2016)

[23] Pearson, K.: The problem of the random walk. Nature **72**(1865), 294–294 (1905)

[24] Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 132–149 (2018)

[25] Miao, Z., Humphreys, B.D., McMahon, A.P., Kim, J.: Multi-omics integration in the age of million single-cell data. Nature Reviews Nephrology **17**(11), 710–724 (2021)

[26] Zappia, L., Phipson, B., Oshlack, A.: Splatter: simulation of single-cell rna sequencing data. Genome biology **18**(1), 1–15 (2017)

[27] Mimitou, E.P., Lareau, C.A., Chen, K.Y., Zorzetto-Fernandes, A.L., Hao, Y., Takeshima, Y., Luo, W., Huang, T.-S., Yeung, B.Z., Papalexi, E., *et al.*: Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. Nature biotechnology **39**(10), 1246–1258 (2021)

[28] Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck III, W.M., Hao, Y., Stoeckius, M., Smibert, P., Satija, R.: Comprehensive integration of single-cell data. Cell **177**(7), 1888–1902 (2019)

[29] Stephenson, E., Reynolds, G., Botting, R.A., Calero-Nieto, F.J., Morgan, M.D., Tuong, Z.K., Bach, K., Sungnak, W., Worlock, K.B., Yoshida, M., *et al.*: Single-cell multi-omics analysis of the immune response in covid-19. Nature medicine **27**(5), 904–916 (2021)

[30] Chen, S., Lake, B.B., Zhang, K.: High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. Nature biotechnology **37**(12), 1452–1457 (2019)

[31] Clark, S.J., Argelaguet, R., Kapourani, C.-A., Stubbs, T.M., Lee, H.J., Alda-Catalinas, C., Krueger, F., Sanguinetti, G., Kelsey, G., Marioni, J.C., *et al.*: scnmt-seq enables joint profiling of chromatin accessibility dna methylation and transcription in single cells. Nature communications **9**(1), 1–9 (2018)

[32] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., *et al.*: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)

[33] Nimse, S.B., Sonawane, M.D., Song, K.-S., Kim, T.: Biomarker detection technologies and future directions. Analyst **141**(3), 740–755 (2016)

[34] Wolf, F.A., Angerer, P., Theis, F.J.: Scanpy: large-scale single-cell gene expression data analysis. Genome biology **19**(1), 1–5 (2018)

[35] Amodio, N., Raimondi, L., Juli, G., Stamato, M.A., Caracciolo, D., Tagliaferri, P., Tassone, P.: Malat1: a druggable long non-coding rna for targeted anti-cancer approaches. Journal of hematology & oncology **11**(1), 1–19 (2018)

[36] Cytlak, U., Resteu, A., Pagan, S., Green, K., Milne, P., Maisuria, S., McDonald, D., Hulme, G., Filby, A., Carpenter, B., *et al.*: Differential irf8 transcription factor requirement defines two pathways of dendritic cell development in humans. Immunity **53**(2), 353–370 (2020)

[37] Pandey, K., Zafar, H.: Inference of cell state transitions and cell fate plasticity from single-cell with margaret. bioRxiv (2021)

[38] Almeida, A.R., Neto, J.L., Cachucho, A., Euzébio, M., Meng, X., Kim, R., Fernandes, M.B., Raposo, B., Oliveira, M.L., Ribeiro, D., *et al.*: Interleukin-7 receptor α mutational activation can initiate precursor b-cell acute lymphoblastic leukemia. Nature communications **12**(1), 1–16 (2021)

[39] Pleshkan, V., Zinov'Eva, M., Vinogradova, T., Sverdlov, E.: Transcription of the klrb1 gene is suppressed in human cancer tissues. Molekuliarnaia Genetika, Mikrobiologiia i Virusologiia (4), 3–7 (2007)

[40] Ng, S.S., De Labastida Rivera, F., Yan, J., Corvino, D., Das, I., Zhang, P., Kuns, R., Chauhan, S.B., Hou, J., Li, X.-Y., *et al.*: The nk cell granule protein nkg7 regulates cytotoxic granule exocytosis and inflammation. Nature immunology **21**(10), 1205–1218 (2020)

[41] Burkhardt, D.B., Stanley, J.S., Tong, A., Perdigoto, A.L., Gigante, S.A., Herold, K.C., Wolf, G., Giraldez, A.J., van Dijk, D., Krishnaswamy, S.: Quantifying the effect of experimental perturbations at single-cell resolution. Nature biotechnology **39**(5), 619–629 (2021)

[42] Moss, P.: The t cell immune response against sars-cov-2. Nature immunology, 1–8 (2022)

[43] André, S., Picard, M., Cezar, R., Roux-Dalvai, F., Alleaume-Butaux, A., Soundaramourty, C., Cruz, A.S., Mendes-Frias, A., Gotti, C., Leclercq, M., et al.: T cell apoptosis characterizes severe covid-19 disease. Cell Death & Differentiation, 1–14 (2022)

[44] Argelaguet, R., Clark, S.J., Mohammed, H., Stapel, L.C., Krueger, C., Kapourani, C.-A., Imaz-Rosshandler, I., Lohoff, T., Xiang, Y., Hanna, C.W., *et al.*: Multi-omics profiling of mouse gastrulation at single-cell resolution. Nature **576**(7787), 487–491 (2019)

[45] Liao, Y., Smyth, G.K., Shi, W.: featurecounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics **30**(7), 923–930 (2014)

[46] Lun, A.T., Bach, K., Marioni, J.C.: Pooling across cells to normalize single-cell rna sequencing data with many zero counts. Genome biology **17**(1), 1–14 (2016)

[47] Unterman, A., Sumida, T.S., Nouri, N., Yan, X., Zhao, A.Y., Gasque, V., Schupp, J.C., Asashima, H., Liu, Y., Cosme, C., *et al.*: Single-cell multi-omics reveals dyssynchrony of the innate and adaptive immune system in progressive covid-19. Nature Communications **13**(1), 1–23 (2022)

[48] Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)

[49] Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487 (2016). PMLR

[50] Melvin, W.J., Audu, C.O., Davis, F.M., Sharma, S.B., Joshi, A., DenDekker, A., Wolf, S., Barrett, E., Mangum, K., Zhou, X., et al.: Coronavirus induces diabetic macrophage-mediated inflammation via setdb2. Proceedings of the National Academy of Sciences **118**(38) (2021)

[51] Sasaki, Y.: The truth of the F-measure. 5 (2007)

[52] Watford, W.T., Hissong, B.D., Durant, L.R., Yamane, H., Muul, L.M., Kanno, Y., Tato, C.M., Ramos, H.L., Berger, A.E., Mielke, L., *et al.*: Tpl2 kinase regulates t cell interferon-γ production and host resistance to toxoplasma gondii. The Journal of experimental medicine **205**(12), 2803–2812 (2008)

[53] Metz, C.E.: Basic principles of roc analysis. In: Seminars in Nuclear Medicine, vol. 8, pp. 283–298 (1978). Elsevier

Appendix

Preprocessing

GSE128639

The expression matrices was used as quantified in the original experiment [52]. For gene expression, standard log-normalization with default parameters in Seurat [28] was conducted. The only difference with the original implement in paper is that we take the raw data of HTO separately from the dataset as the third omics. HTO is an extremely sparse data so that with this as a third omics, the performance of Seurat 4.0 will be strongly lagged back.

GSE156478

The control and stimulated CITE-seq were filtered based on the following criteria: mitochondrial reads greater than 10%; the number of expressed genes less than 500; the total number of UMI less than 1000; the total number of ADTs from the rat isotype control greater than 55 and 65 in the control and stimulated conditions respectively; the total number of UMI greater than 12,000 and 20,000 for the control and stimulated conditions respectively; the total number of ADTs less than 10,000 and 30,000 for control and stimulated conditions respectively. The cells that were classified as doublets in the original study were filtered out. For the ASAP-seq data, cells with a number ADTs more than 10,000 and number of peaks more than 100,000 were filtered out. Finally, 4502 cells (control) and 5468 cells (stimulated) from ASAP-seq, 4644 cells (control), and 3474 cells (stimulated) from CITE-seq were included in the downstream analysis. The number of common genes across the four matrices is 17441 and the number of common ADTs is 227 [16].

scNMT

Gene counts were quantified from the mapped reads by featureCounts [45], and gene annotations were pbtained from Ensembl version 87 [46]. Only protein-coding genes mathcing canonical chromosomes were considered. For methylation and accessibility pseudo-bulk profiles, the values were averaged using running windows of 50 bp. The information from multiple cells was combined by calculating the mean and the standard deviation for each running window. Accessibility profiles were processed with each cell and gene in +/- 200 bp windows around the TSS. Only genes covered in at least 40% of the cells with a minimum coverage of 10 GpC sites were considered [31].

SNARE

SNAREseq [30] consists of chromatin accessibility and gene expression. The data is collected from a mixture of human cell lines: BJ, H1, K562, and GM12878. We reduce the dimension of the data by PCA. The size of the resulting matrix for scATAC-seq is of 1047×1000 and 1047×500 for the gene matrix. We use the code provided by the author to generate annotations for BJ, H1, K562, and GM12878.

COVID-PBMC

We mostly follow the preprocessing of the original paper as [47]. Briefly, FASTQ files were generated from raw sequencing reads by Cell Ranger mkfastq pipeline. Cell Ranger count pipeline (v3.1) was utilized to perform alignment, filtering. barcode counting, and UNI counting. GRCh38 was denoted as genome reference. To remove dead and dying cells, Cells with mitochondrial gene percentages higher than 12% and cells with less than 200 genes was filtered out. For CITE-seq samples, the cells were demultiplexed and hashing adt COUNTS were removed. The remaining counts were normalized by library size and square. For TCR data, the raw sequencing reads of the T cell receptor (TCR) libraries were prcessed by the Cell Ranger V(D)J pipeline by 10x Genomics. Only V(D)J contigs with high confidence defined by cell ranger were considered. The cells of one beta chain contig and zero or one alpha chain contig were remained [47].

Evaluation metrics

ACC We denote *Positive* as P, *Negative* as N, *True positive* as TP, *False negative* as FN, *False positive* as FP, and *True negative* as TN. Then we can define accuracy (ACC) [53] as

$$ACC = \frac{TP + TN}{P + N}. \tag{6}$$

F1 score Here, F1 score can be calculated [51] as:

$$F1\ score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}. \tag{7}$$

The F1-macro is the arithmetic mean of all the per-class F1 scores, and F1-weighted is computed by taking the mean of all per-class F1 scores considering the weight. Weight refers to the number of actual occurrences of the class in the dataset.

ARI Adjusted rand index (ARI) is used to measure the similarity between the predicted labels and ground truth. The Rand Index (RI) calculates a similarity measure between two clusterings, taking all pairs of samples into consideration. It counts pairs that are assigned in the same or different clusters in the predicted and actual clusterings. ARI is a corrected-for-chance version of the Rand index defined as

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max[\mathrm{RI}] - \mathbb{E}[\mathrm{RI}]} \tag{8}$$

where $\mathrm{RI} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN++FP++FN}}$ and $\mathbb{E}$ is the expectation value.

10.1 The results of parameter sensitivity show the necessity for omics-transition ability enabled by omics2vec

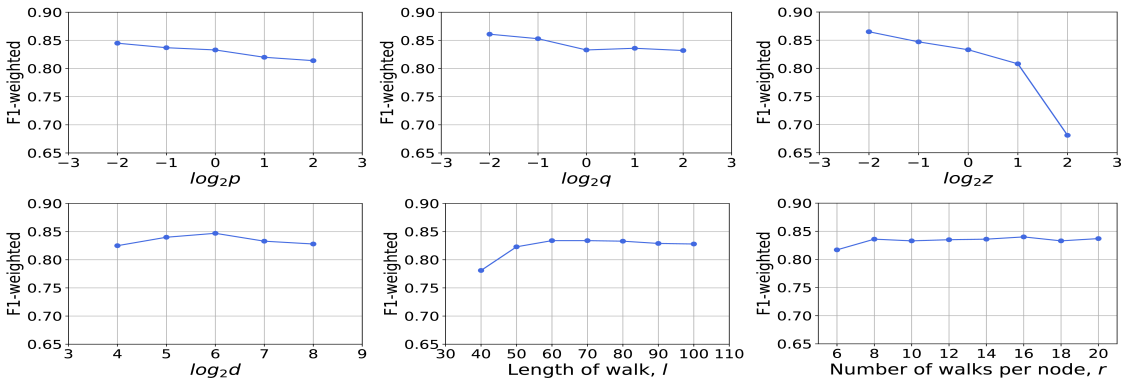

Figure 7: Parameter sensitivity on omics2vec. During the examination of one parameter, our parameters are all set to the default value.

The omics2vec algorithm involves a number of parameters and in Figure 7, we examine how the combination of parameters influences the performance of this algorithm. We test the classification performance on the SNARE dataset using the 10% label for fine-tuning. Except for the parameter being tested, all other parameters are set as the default values. During the examining, we repeat the experiment 20 times with different random seeds and take the average of the results. The default values for p, q, and z are all the unity. The default value for embedding dimension d is 128, for the length of walks l is 80 and for the number of walks per node r is 10.

We measure the F1-score as a function of parameters. The performance of omics2vec is not very sensitive to the in-out parameter p and the return parameter q. It shows that the performance slightly improves as p and q decrease. This might be caused by the homophilic of the dataset. However, the jump parameter z greatly improves the performance of omics2vec when it decreases. The improvement can be based on the better connectivity among different omics we expect to achieve from the motivation of omics2vec. While a low z encourages the jump from one omics to the others, it efficiently grabs valid neighborhood information from other omics.

We also examine how the number of features d and the neighborhood parameters of the node, *i.e.*, number of walks r and walk length l, affect the performance. We observe that the performance tends to saturate once the dimensions of the embedding go around 2^6. Moreover, increasing the length of walks

l and the number of walks per node r also improve the performance which is caused by the enlarged sampling budget.

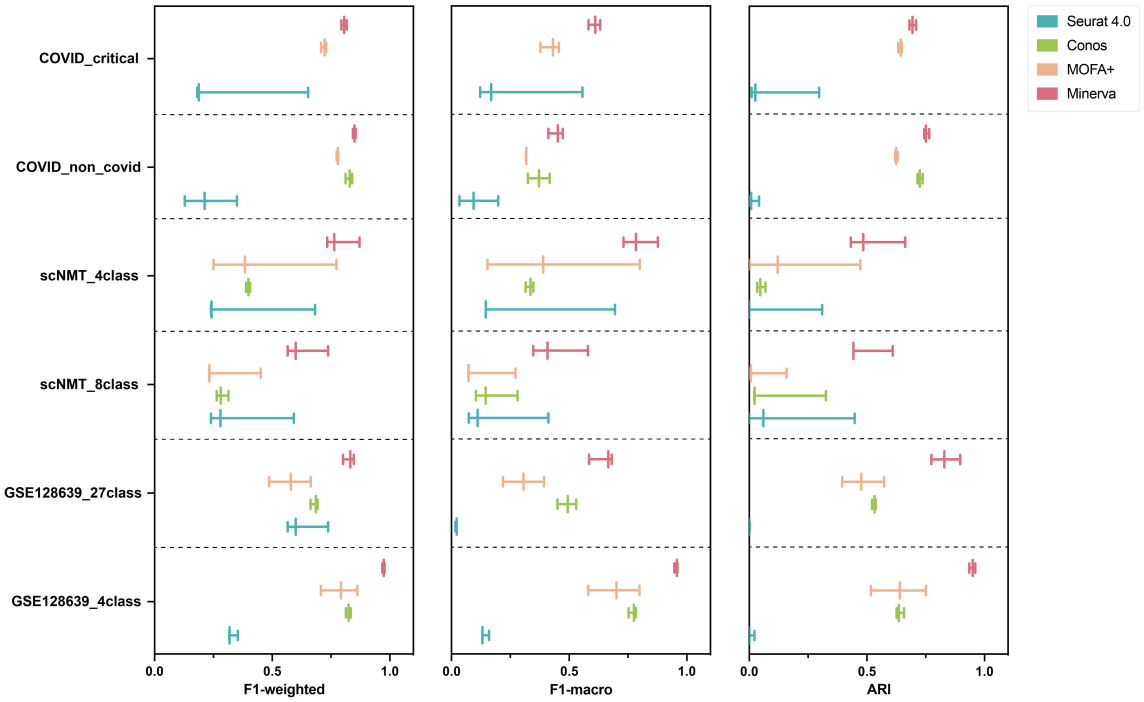

Figure 8: Comparison of classification F1-weighted score, F1-macro score and ARI on four datasets with six set of annotations. Each row contains three ticks which represent a method's performance on a dataset with test size 95%, 90% and 80% respectively.

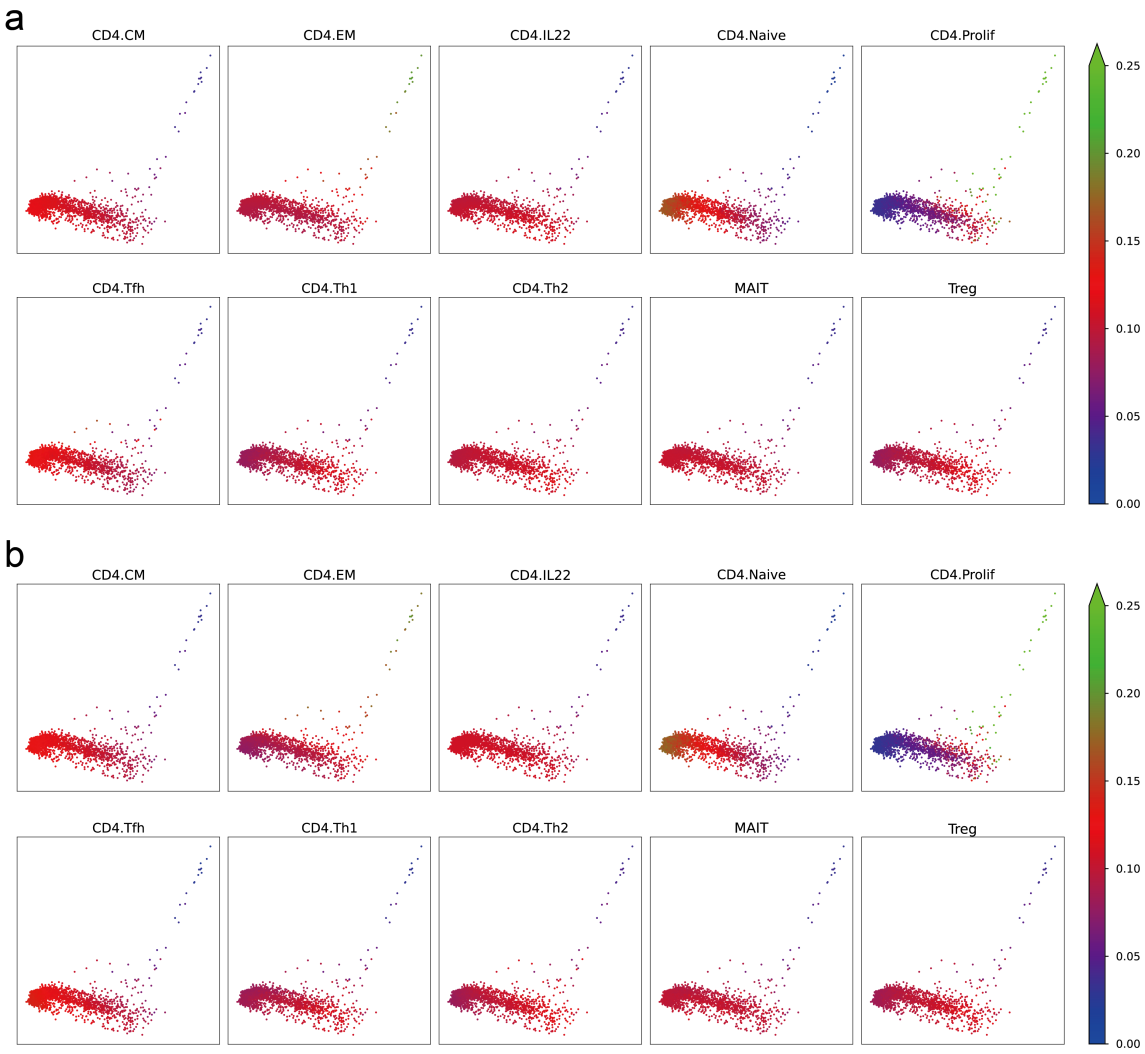

Figure 9: The cell differentiation tendency analysis on healthy cells. **a**. The differentiation score on CD4 Naive cells to different cell types inferred from ground-truth label. **b**. Same as **a** but is interred from scMinerva's predicted label. In all the cell types, our method shows a strong approximation to the annotation's result.

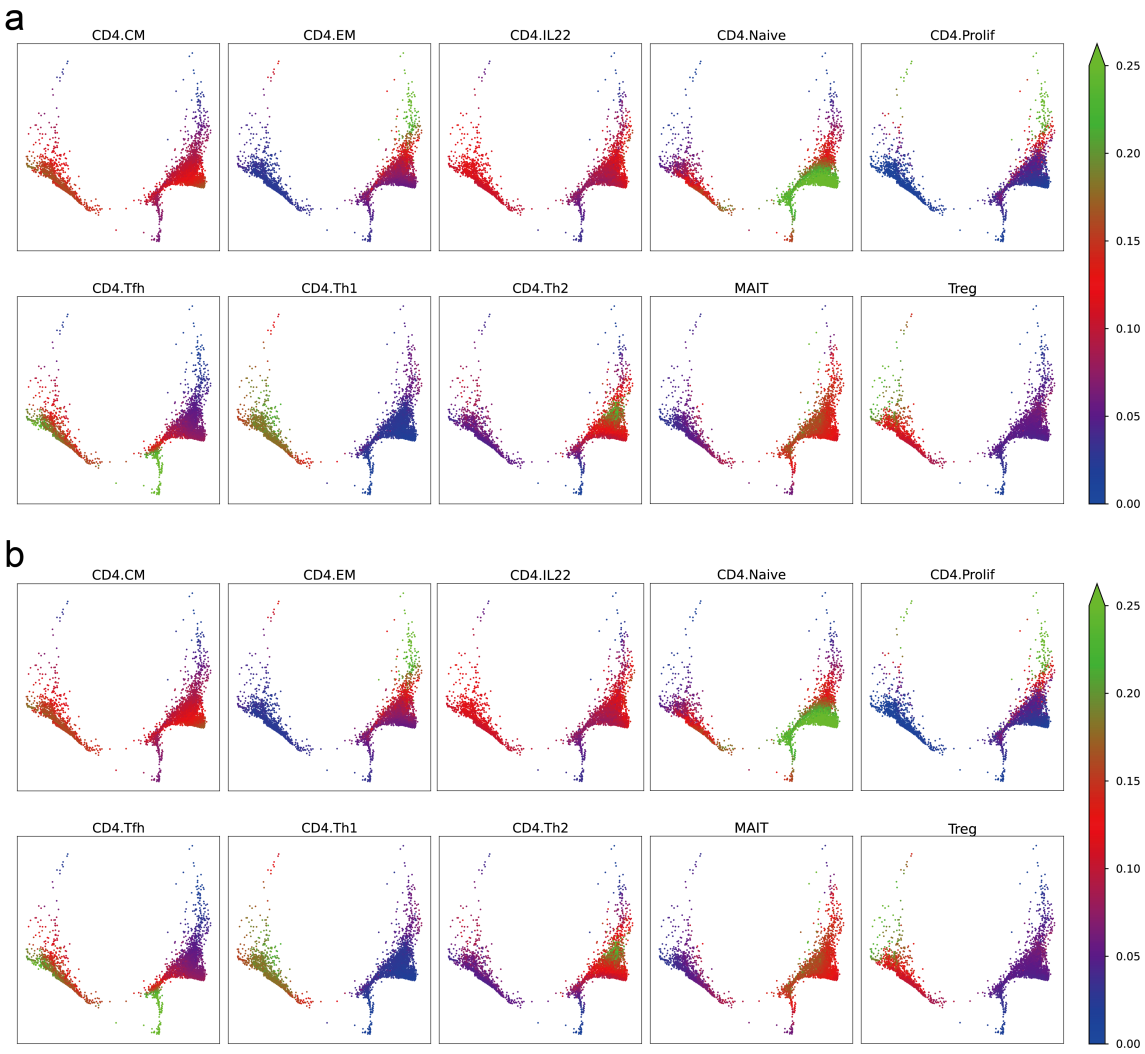

Figure 10: The cell differentiation tendency analysis on infected cells. **a**. The differentiation score on CD4 Naive cells to different cell types inferred from ground-truth label. **b**. Same as **a** but is interred from scMinerva's predicted label. In all the cell types, our method shows a strong approximation to the annotation's result.