scMinerva: a GCN-featured Interpretable Framework for Single-cell Multi-omics Integration with Random Walk on Heterogeneous Graph

Tingyang YU [2] [3], Yongshuo Zong [4], Yixuan Wang [1] [5], Xuesong Wang [1], and Yu Li [*1] [6]

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China

[2]Department of Mathematics, The Chinese University of Hong Kong, Hong Kong SAR, China

[3]Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China

[4]School of Informatics, The University of Edinburgh, Edinburgh, United Kingdom

[5]Department of Mathematics, Harbin Institute of Technology, Weihai, China

[6]The CUHK Shenzhen Research Institute, Hi-Tech Park, Nanshan, Shenzhen, China

Abstract: The development of single-cell multi-omics technologies profiles DNA, mRNA, and proteins at a single-cell resolution. To meet the demand, we present **scMinerva** for single-cell multi-omics integration utilizing graph convolutional networks and a new random walk strategy, which outperforms existing methods on various datasets. Our method is especially robust on high-noise more-omics data and is lightweight concerning speed and memory. scMinerva can effectively perform downstream tasks, such as biomarker detection and cell differentiation analysis. We extensively interpret the robustness of scMinerva by analyzing components' occurrence frequency in walks during training at omics level, cell-type level, and single-cell level.

Keywords: *single cell, multi-omics integration, interpretability, interpretability, graph convolution network, random walk*

Background

Many single-cell technologies are now developed to measure the biological systems, revealing a significant heterogeneity across specific cell types and cell states. Starting from the transcriptome analysis [1], such as RNA sequencing or RNA micro-array, the measurement tools have rapidly been extended to multi-omics in recent years, such as protein or DNA content [2]. This trend greatly enhances the biological discoveries at the single-cell level and can provide a more comprehensive molecular mechanism by integrating multi-omics data.

However, the inherent characteristics of single-cell multi-omics data make it difficult to analyze. First, single-cell data is extremely sparse due to the lack of expression of genes in specific cell types and the relatively shallow sequencing of some droplet-based technologies [3]. Second, the data is highly noisy due to current technical limitations, including amplification bias, low capture rate, dropout, *etc* [4]. Also, the high-dimension features produced by the multiplexing and throughput of multi-omics data lead to remarkably expensive computational overhead.

[*] Email of the corresponding author: liyu@cse.cuhk.edu.hk

To address these problems, some methods have been proposed for single-cell multi-omics data integration. They can be classified into three categories:

1) Latent-space inference methods. This type of methods regards each omics as a view of the underlying relationship of cells and assumes a common latent space shared by all the omics. Latent space approaches target at the feature level by matrix factorization and manifold alignment. Typical methods include MOFA [5] , scAI [6], and MOFA+ [7]. However, matrix factorization methods ([5]) are under a linear operation that does not suit the characteristics of the data sparsity, while the assumption of Manifold alignment methods ([6] - [8]) that requires the globally matched distributions are often too strong, as different omics are derived from different tissues and cell types.

2) Correlation-based methods. Such correlation-based methods focus on the (dis)similarity measures that correlate different components to each other. In this setting, for example, Seurat 4.0 [9] utilized weighted nearest neighbor analysis. CiteFuse [10] implemented the similarity network fusion, and Conos [11] used the graph-correlation to identify nearby cells. But these methods ([9][11]) designed for unpaired data require a separate feature selection step before integration for dimension reduction, and the performance is sensitive to the genes selected. Also, similarity measurement is especially memory-consuming and thus not scalable to gigantic single cells datasets.

3) Deep learning-based methods. These approaches utilize the capability and flexibility of Neural Networks to model complex data. Some of these methods are unsupervised such as TotalVI [8] and DeepMAPS [12], while there also exist some semi-supervised or supervised methods, like scJoint [13]. But most existing works are based on Convolutional Neural Networks (CNN), which did not consider nodes' spatial information and failed to capture the correlations between cells effectively.

Besides, there are also some common limitations shared by the existing methods. 1) Cannot take advantage from different omics when dealing with high-noise more-omics datasets (*i.e.*, three-omics). Most of the existing methods are only designed for datasets with two omics and cannot fully utilize the other ones, *e.g.* Seurat 4.0 [9]. Forcibly integrating other omics cannot benefit the result and even impair the performance. 2) High memory consumption. Many existing methods, like CiteFuse [10], have very expensive memory demands. The computational complexity is not affordable for users with standard devices. 3) Lack of interpretability. This is especially common in existing deep learning methods. Some of them straightly apply neural networks to the framework which mostly produce a "black-box" model and their functionalities hardly form biological meaning. The lack of interpretability obstructs their wider usage in realistic settings.

To address these limitations, we propose scMinerva, a more flexible multi-omics integration framework that can adapt to any number of omics with efficient computational consumption. Considering the structure and biological insight of this multi-omics integration problem, to learn the cell property on top of multi-omics information and the cell neighbors, we accordingly design the model on a new random walk strategy. It allows our framework to process any number of omics and has an explicit probabilistic interpretability, and a Graph Convolutional Network (GCN), which considers the spatial information of nodes and endows the method a strong robustness to noises.

Specifically, 1) to effectively take advantage from any number of omics, we construct a heterogeneous graph among all the omics and implement GCN to strengthen the model's robustness. We first build topology for each omic as a sub-graph and then link nodes from the same cell across all the omics. By our new random walk strategy, we replenish our knowledge space to the neighbors of one cell concerning all the available omics. The GCN will jointly benefit the model by adjusting the weights of omics-transition links to eliminate noises from different omics and reduce the negative influence from data sparsity. Our framework therefore shows an remarkable robustness on high-noise more-omics datasets.

2) scMinerva is also computationally efficient. Our new walking strategy is a variant of node2vec [14]. Node2vec is a combination of random walk and word2vec model [15]. For our new walking strategy, instead of combining Deep-First Search and Breadth-First Search as node2vec, we add another dimension as "Omics-First Search", which guarantees the visit to its mapping nodes on the sibling omics. On the one side, as a variation of it, our method shares a fairly low computational efficiency as node2vec in terms of both space and time requirements [14]. On the other side, our method only trains the GCN to obtain the weights of omics-transition links with a linear time complexity in terms of the number of cells.

3) scMinerva is designed from the start to be more interpretable. Since GCN is hard to be straightly interpreted, we implement random walk which is a transparent probabilistic model and word2vec [15] which is better-studied under maximum likelihood estimation. They jointly build a possible "window" to the "black-box" of GCN so that we can semi-transparently learn how GCN functions on our problem. In summary, we analyze the changes on generated walks during training from the perspective of word2vec's

mechanism. These changes indicate that GCN effectively integrates information from different omics by providing a more reasonable omics-transition probability for random walk. Moreover, by evaluating the occurrence frequency of nodes in walks, we found that GCN overcomes the problem of the data sparsity by assigning higher weights to nodes with high feature expression level. It reveals a potential on detecting high gene expression cells and justifies scMinerva's output as they are formed from anchoring representatives in cell types.

To summarize, our main contributions are

- we present an unsupervised integration method scMinerva for single-cell multi-omics data. It is flexible to integrate any number of omics and scalable to large-scale single-cell datasets with low time complexity and memory consumption.
- we extensively evaluate the proposed method among a wide range of data and downstream tasks, which proves its effectiveness and robustness, especially in high-noise more-omics cases.
- we interpret the effectiveness of GCN on handling the sparsity and high noises of single-cell data. Specifically, we analyze the changes on generated walks during training under the mechanism of word2vec and further conclude scMinerva's potential on detecting high gene expression cells.

Result

scMinerva is designed for multi-omics intergration problem

In this study, we introduce scMinerva, an unsupervised Single-Cell Multi-omics INtegration method with GCN on hEterogeneous graph utilizing RandomWAlk. Its framework is shown in Figure 1. In single-cell multi-omics data, each cell has its expression vectors on different omics which measure different aspects of its biological process. Accordingly, if we solely build topology for each omics, a cell will have a mapping node on each of them. Inspired by this, scMinerva is designed to generate latent embeddings for cells by integrating the information of their neighbors across all the omics. For an easier discussion, we assume there is an "explorer" on the graph who is integrating information about neighbors of a cell. Intuitively, when the explorer is trying to learn the neighbors' information of the cell's mapping node in one omics, we allow it "jump" to the cell's mapping nodes in other omics to build better insight into that cell stage. Below we introduce each step in detail.

We formulate the integration as a graph learning problem where graph nodes are heterogeneously from different omics. After preprocessing, we first separately build a weighted K-nearest neighbor (KNN) graph for each omics by their Euclidean similarity matrix and name them as sub-graphs. Then, we build a heterogeneous graph by linking the mapping node of a cell in one sub-graph with its mapping nodes in the other sub-graphs. To remark, the graph is directed and therefore the transition probability from node A to node B might differ from its opposite.

With the heterogeneous graph, we develop a new walking strategy to fit this biological problem. It can be regarded as a promoted version of node2vec [14]. Briefly, node2vec learns the graph topology by random walk and introduces two parameters named p and q, where p controls the likelihood of immediately revisiting a node in the walk and q allows the search to differentiate between "inward" and "outward" nodes [14]. In our strategy, in addition to providing p and q, we introduce a third parameter z to control an inter-omics transition within the frame. Hence, the explorer is not restricted from the partial information in one omics but can learn neighbors of nodes from the same cell in all the omics. These parameters define the transition probability and complete the set-up for random walk.

Then, we input walks into the word2vec model and obtain the node embeddings of the heterogeneous graph. Notably, word2vec will form embeddings to ensure the biologically similar cells to have a small distance in the embedding space. And to optimize the embeddings, we input them and the heterogeneous graph's topology (*i.e*, edge list) into the GCN model. Here, we utilize the loss function from DeepCluster which disperses different clusters detected by K-means [16] and GCN with output the trained omics-transition weight matrix. After the training, we decode the nodes in walks from an omics-specific manner to the original cells and generate embeddings by word2vec in a low-dimension space for downstream works. The training details of the GCN model can be referred in Method Model Training.

The embeddings are used to classify different cell types and produce predicted labels. With the predictions, we can perform multiple downstream tasks, such as detecting marker genes in various cell

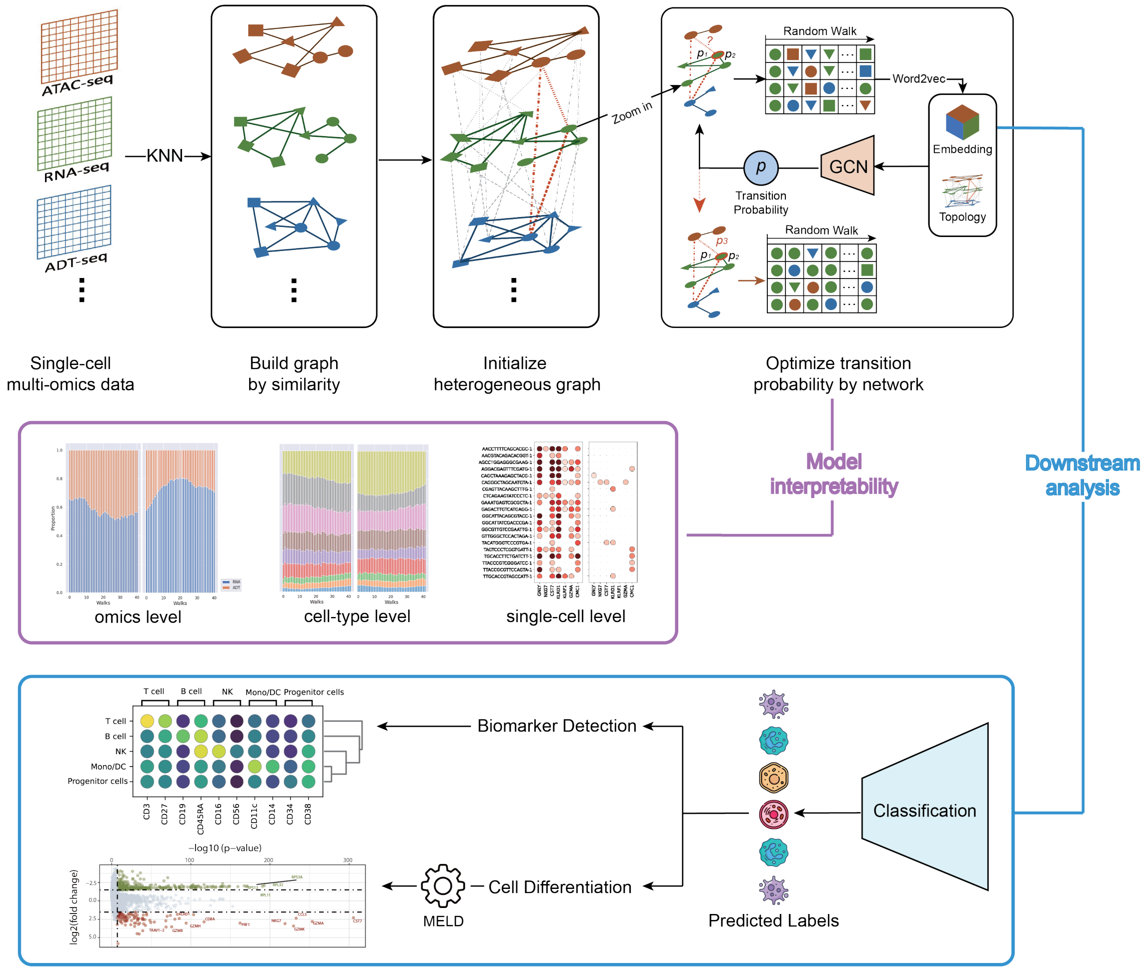

Figura 1: Framework of scMinerva. scMinerva is a method designed for single-cell multi-omics integration and is adaptive to any number of omics. It formulates the setting as a graph learning problem, solely builds topology for each omics and learns the graph structure on a reconstructed heterogeneous graph. The main framework can be divided into two parts: random walk and Graph Convolutional Network(GCN) training. In the random walk phase, we provide a new walking strategy to allow an inter-omics transition and obtain embedding by inputing walks into word2vec's model. In the training phase, we input the embeddings and the heterogeneous graph's topology into the GCN model to search for an optimal transition probability which contributes to the graph in the next epoch to achieve a better random walk. After training, the learned embeddings are validated in various downstream tasks, such as label classification, biomarker detection, cell differentiation analysis, *etc.*. We also interpret GCN's effectiveness by evaluating the changes on walks during training at omics level, cell-type level and single-cell level.

states and analyzing potential cell differentiation in a single-cell resolution. More interestingly, with the help of random walk and the probabilistic property of word2vec model, we interpret reasons of the robustness of scMinerva and how GCN functions in the framework. By analyzing the occurrence frequency of the component of the walks at omics level and cell-type level, we conclude that GCN benefits the model by providing a more reasonable omics-transition probability. Furthermore, at the single-cell level, we observe that the output of the framework relies more on nodes with higher feature expression level. This explains the effectiveness of our model on sparse data and reveals the potential for detecting high feature expression cells.

scMinerva shows a linear time complexity and low memory consumption

Our method is computationally efficient in terms of both space and time requirements. The main features of scMinerva are random walk and a GCN model. For an easier demonstration later, assume the dataset has n samples and c omics. Denote the constructed heterogeneous graph as $\mathcal{G}$. And $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the node set and $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V}, \mathbf{R})$ is the weighted edge set.

For random walk, the space complexity of every node in the graph is $O(|\mathcal{E}|)$ which only stores the immediate neighbors of the node. Based on our walking strategy, the probabilistic model is a 2^{nd}-order Markov chain. It is necessary to store the connecting edges as well as the transition probability between the neighbors of a node and itself. Thus, the space complexity is $O(a^2|\mathcal{V}|)$, where a is the average degree of the graph. The heterogeneous graph is built by the KNN algorithm with added omics-to-omics links on every node. Suppose we choose k neighbors during the construction on KNN, the nodes of the heterogeneous graph will have an average degree of $k + c - 1$. Therefore, a is bounded as a reasonably low constant. Overall, the space complexity of random walk is $O(|\mathcal{E}| + (k + c - 1)^2|\mathcal{V}|)$. Furthermore, considering the time complexity of the strategy, since we impose the graph connectivity in the general process, the sampled walks are reusable across different source nodes. Assume each random walk is of length l, for $w < l$, we can generate w samples for $l - w$ nodes from its Markovian nature. Therefore, the complexity on time is $O(\frac{l}{w(l-w)})$ for each sample.

For the GCN, its efficientness on time complexity and space complexity is broadly studied [17]. To further reduce the complexity, the proposed method does not need to train all the edge weights, which are of complexity $O(n^2)$. Instead, we only require GCN to train the omics-to-omics links in the heterogeneous graph. Therefore, the output is of size $O(n)$, and the coefficient of complexity is $c(c+1)$. Since the number of omics c is always small in practice, we greatly reduce the requirement on complexity to a linear case to achieve efficient computation.

scMinerva has an impressive anti-noise ability on simulated four-omics datasets

Tabela 1: Performance comparison on simulated data where the first column gives number of samples in datasets. Bold indicates the best method, and the underline indicates the second-best method. scMinerva outperforms the second-best method by around 30% across all the metrics.

#Sample	Method	ACC	F1-weighted	F1-macro	ARI
2k	**scMinerva**	**0.911**	**0.913**	**0.913**	**0.781**
	MOFA+	0.474	0.464	0.462	0.127
	Conos	0.627	0.616	0.615	0.307
	Seurat 4.0	0.263	0.209	0.211	0.004
5k	**scMinerva**	**0.972**	**0.972**	**0.972**	**0.913**
	MOFA+	0.547	0.548	0.548	0.198
	Conos	0.764	0.764	0.656	0.449
	Seurat 4.0	0.411	0.408	0.407	0.099
10k	**scMinerva**	**0.957**	**0.957**	**0.957**	**0.895**
	MOFA+	0.691	0.686	0.686	0.41
	Conos	0.719	0.714	0.714	0.450
	Seurat 4.0	0.220	0.165	0.164	0.001
30k	**scMinerva**	**0.978**	**0.978**	**0.978**	**0.947**
	MOFA+	0.675	0.676	0.677	0.377
	Conos	0.447	0.446	0.446	0.134
	Seurat 4.0	0.256	0.230	0.230	0.007

Initially, we evaluate our method on simulated datasets. As far as we know, among all the existing real-world datasets, there is currently no publicly available four-omics single-cell data. However, it is a common agreement that there is a rapid trend in the development of experimental methods which jointly profile three or more omics [18]. And consequently, more flexible computational algorithms for more-omics data will be greatly required to adapt to this phenomenon. Therefore, we validate scMinerva using simulated four-omics data. It was produced by a single-cell RNA (scRNA) data simulator, splatter [19]. We took the RNA-seq of GSE156478-CITE [20] as an input to obtain simulated RNA-seq. To learn realistic mappings, we trained neural networks with real-world datasets as the input. Three networks were trained by mapping GSE156478-CITE RNA-seq to its ADT, GSE156478-ASAP ATAC-seq to its ADT, and sci-CAR RNA-seq to its ATAC-seq. By inputting the simulated RNA-seq to the above networks, we generate four-omics datasets with 5 classes and sample number 2k, 5k, 10k, and 30k, respectively. The simulated RNA-seq, ATAC-seq, ADT from RNA-seq, and ADT from ATAC-seq data are of feature number 815, 2613, 227, and 227 respectively. Full details on data simulation can be referred to Method Data Simulation.

The results are listed in Table 1, where we compared the performance with Seurat 4.0 [9], MOFA+ [7], and Conos [11]. To note, TotalVI [8], CiteFuse [10], and DeepMAPS [12] are not capable of processing more than two omics and are thus not included here. But they will be compared in the later section on 2-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 datasets, our method shows around 30% improvements over the second-best method.

As we have demonstrated before, most of the existing methods cannot fully take advantage of all the omics when encountering more-omics datasets(*i.e.* more than two omics). Therefore, they are strongly lagged back by the low-quality omics in datasets and perform terribly. The huge gap between scMinerva and other existing methods comes from the differences in the anti-noise ability. scMinerva guarantees robustness on anti-noise as it is not sensitively affected by the low-quality omics in datasets.

After extensively interpreting the model, we find that our model's anti-noise ability comes from taking advantage of the more reliable omics in a cell-specific manner. In general, for each cell, the difficulty level to correctly classify it is different under different omics. We name the omics that can easily perform a correct classification on some cell as a "high-quality" omics to this cell while the opposite is a "low-quality" omics to it. Intuitively, when learning the neighbors of this cell on the heterogeneous graph, GCN will assign a higher transition probability from its mapping node on the low-quality omics to the counterpart on the high-quality omics and lower the transition probability of its opposite. So we can mostly learn from the cell's neighbors on the sides which are easier for classification and diminish the negative influence of low-quality omics. This observation comes from interpreting GCN by analyzing the changes in walks' components under the mechanism of the word2vec model. We will further demonstrate it in the later sections.

scMinerva achieves state-of-the-art performance on six real-world datasets over five existing methods

Furthermore, we evaluate our method on real-world datasets. Their sample sizes range from 1k to 64k. We run the experiments on CITE-seq (GSE128639 [21], GSE156478-CITE [20], COVID-PBMC [22]), ASAP-seq (GSE156478-ASAP [20]), SNARE-seq [23], and scNMT-seq [24]. For the COVID-PBMC dataset, we take the healthy samples and critical symptom samples separately, and form the COVID-non-covid dataset and 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). 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 [25]. The details for data preprocessing are listed

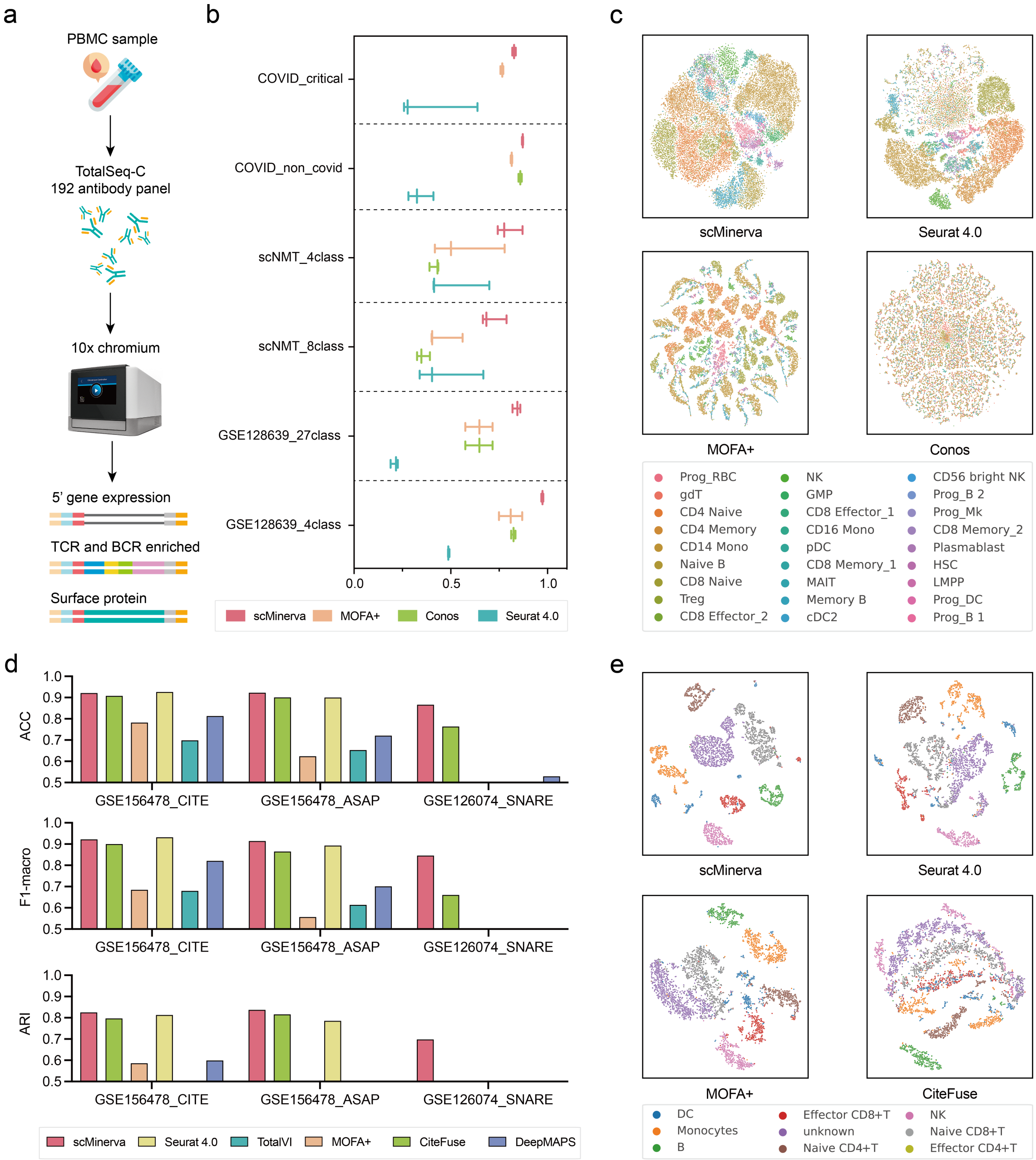

Figura 2: Classification performance on real-world datasets. **a**. The sequencing technique for Peripheral blood mononuclear cells to obtain three-omics data. **b**. Classification accuracy comparison on four three-omics datasets with six sets of annotations. Each row contains three ticks which represent a method's performance on a dataset with test sizes 95%, 90%, and 80% from left to right, respectively. Our method outperforms all the experiments and is not sensitive to the test size compared to other methods. **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. 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.

in Appendix Preprocessing.

As annotating single-cell data is extremely labor-intensive and time-consuming, it is expected that the method can still be effective with only a few ground-truth labels for classification. Therefore, we fit the KNN classifier using the annotated data with the sizes of only 5%, 10%, and 20% of the whole training set to evaluate the performance. For full details on classification, please refer to Method Classification. 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.

We list the performance on accuracy of different methods in Figure 2b, and the performance on other three metrics in Appendix 7. We observe that scMinerva outperforms the compared integration methods on all the classification tasks of all test proportions. As a remark, Conos failed to work in COVID-critical datasets with a very sparse omics matrix obtained by Hash Tag Oligonucleotide (HTO) [26]. Notably, when using annotations with 10% of the data size as the training set, scMinerva improves the classification accuracy by 7% on average and up to 20% on GSE128639 when classifies to 27 classes compared with the second-best method. Also, it shows the stability when only 5% labels were used for training. From Figure 2**b**, it can be easily observed that our method not only has the best performance but also has the least variance compared with others. The power of scMinerva is sparkled by easy fine-tuning and is not sensitive to the size of the training set.

We also visualize the embeddings of the GSE128639 dataset produced by scMinerva, Conos, Seurat 4.0, and MOFA+ via t-SNE [27] in Figure 2**c**. GSE128639 has 30k cells across 27 cell types and cell states. From the scatter colored by the ground truth labels, it can be seen that scMinerva clusters samples from the same type together and shows clear boundaries between different types. MOFA+ shows a disperse gathering for samples that are supposed to be in the same cluster, while in the visualizations of Seurat 4.0 and Conos, the embeddings of different cell types overlapped, showing that Seurat 4.0 and Conos are not able to effectively discriminate different cell types.

Moreover, in Figure 2**d**, we list the performance of methods on two-omics datasets. To better show the differences, we start the y-axis from 0.5 and omit bars below this value. GSE156478-CITE and GSE156478-ASAP are two-omics PBMC datasets whose omics are of relatively high quality. As they are easier for classification, most of the existing methods perform well on them. From the results in Figure 2**d**, scMinerva only achieves a slightly better performance on these two datasets than the second-best method. For the other two methods that achieve similar performance with scMinerva, Seurat 4.0 and CiteFuse, CiteFuse is not adaptive to datasets containing more than two omics while Seurat 4.0 also shows a poor performance on the high-noise dataset, such as SNARE. The embedding of GSE156478-ASAP are visualized in Figure 2**e**, colored by ground-truth labels. scMinerva, Seurat 4.0, MOFA+, and CiteFuse are listed. The embedding of scMinerva is clearer than the other three methods as shown in the plot.

However, in another 2 omics dataset, SNARE, scMinerva strongly out-performs all of these existing methods on classification with an around 15% promotion. This performance gap occurs for the dataset with high noise caused by the excellent anti-noise ability of scMinerva. It can effectively handle situations where there is a severe quality gap between different omics. In this case, from the result listed below in Figure 3, we can observe a severe quality gap between two omics of the SNARE dataset, as the performance of ATAC-seq is very poor. However, our method is not affected by the poor quality of certain omics and shows strong robustness. But for all the other methods, including TotalVI, DeepMAPS, Seurat 4.0, and MOFA+, they tend to mess up all samples to one class, while CiteFuse is strongly encumbered by the high noise in ATAC-seq. scMinerva achieves an accuracy of over 80% and has a 20% improvement on ARI over other methods.

The excellent performance of scMinerva compared to existing methods is because it is specially designed for single-cell data that is highly noisy and sparse. In the last section, we discussed how GCN functions on high-noise datasets. So here, we will further give an intuition on how GCN functions on data sparsity.

It is mentioned in the background that single-cell data is extremely sparse due to the low expression level and some artifact dropout. Therefore, the valid information in the matrix as known as the high expression level cells is more than important for data integration in this case. We have observed that, with our random walk strategy, after training, nodes with higher feature expression levels have a higher chance to be included in a walk. This discovery comes from analyzing nodes' occurrence frequency during training and is especially important for the integration of sparse data. GCN assigns a higher "attention" to those more "valuable" cells with rich gene expression. Consequently, we can make full use of their expression information as an anchor when generating walks and embeddings.

This fact strengthens our method's robustness by fully taking advantage of those nodes that are less affected by dropout as well as data sparsity.

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

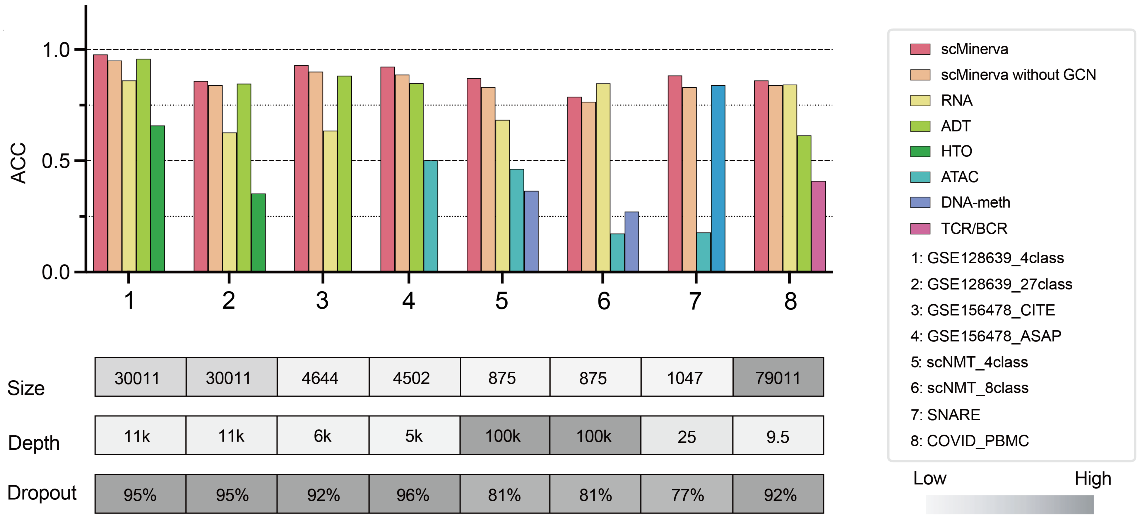

Figura 3: Ablation study results on real-world datasets. The clustered histograms represent results from the same datasets, and below them, we list their sample sizes, measurement depths, and dropout rates. It shows that scMinerva achieves the best performance under most of the cases and efficiently 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.

To validate the robustness of integrating different omics, 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 random walk without the omics-transition parameter, generate embeddings, and fit a KNN classifier for classification. To enable a fair comparison, the data split across all omics 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 the valid information from different omics and obtains a more comprehensive inference.

As we demonstrated before, in practice, we found that it is the GCN model that greatly benefits the framework and strengthens its robustness. Our framework's robustness is built on the following two aspects:

1) For different cells, GCN will effectively increase the probability of walking on its high-quality omics and reduce the chance of walking on its low-quality omics.

2) GCN will weigh nodes with higher feature expression levels heavier for the later random walk and help to fully utilize information from these nodes that are not sparse and less influenced by dropout.

Therefore, in the following two sections, we will illustrate how we conclude the above observations and provide an extensive insight into scMinerva's interpretability.

The component proportion changes at omics level and cell-type level interpret GCN's effectiveness on anti-noise ability

In this section, we investigate the reason behind the performance: why does integrating different omics result in better performance, and how scMinerva integrate them.

In general, single-omics data has limited information compared to multi-omics data as they cannot fully reflect the biological system which is a collaboration among different omics. Take the natural killer cell (NK cell) as an example. NK cell has some special surface proteins among immunocytes that are broadly expressed in most NK cells including immature NK cells (iNK) and mature NK cells (mNK) [28]. So in the real world, researchers tend to weigh surface protein (*i.e.* ADT) higher when distinguishing NK cells from other cell types [29]. However, some initial stage immature NK cells or NK-cell precursors (NKP) are short of typical NK-cell protein markers. This shortage will further mislead the classification results without information from other omics.

Under the above general perspective, we conclude that scMinerva's strong robustness shown in the classification experiments is based on bridging the information gap among different omics. The gap might be caused by artifact error or gene expression lags. But with the GCN, we can effectively mitigate the gap and lead to a more comprehensive insight. Our experiment is as the follows: we analyze 428 NK cells' differentiation tendency on dataset GSE156478 [20], take out cells that are short of typical NK-cell protein markers, and observe whether scMinerva can correctly classify them with the help of other omics and how it performs on these cells during training.

We first analyze the NK cells' ADT raw matrix and calculate their library size which is the total sum of counts across all the ADT features. Then, we cluster these cells by fitting the distribution of normalized library size with Gaussian Mixture Model (GMM). Since NK cells can be generally classified as NKP, iNK, and mNK cells, the number of components of GMM is set to be 3. After fitting the model, in Figure 4**a**, the scatter visualizes NK cells by the cluster they belong to. We can observe that most of the cells are in cluster 1 and cluster 2 which is the main body of NK cells, but some dispersed cells are classified in a different cluster, cluster 0. Also, the histogram shows the normalized library size of these NK cells. We can find that cells from cluster 1 and cluster 2 are of reasonably high library size. However, cells from cluster 0 are of a low library size which means their ADT expression is more sparse than the average of NK cells.

To further figure out what kind of surface proteins cluster 0 is short of, we draw the volcano plot between cells from cluster 0 and cluster 2, which has the highest library size, and run gene enrichment analysis on those highly upregulated genes as Figure 4**b**. To remark, the volcano plot shows the statistical significance (P-value) versus the magnitude of change (fold change). It can reflect the down/up-regulated genes between different groups of cells. We take the cutoff P-value as 1e-7 and the cut-off logarithm fold change as $\frac{3}{2}$. From the volcano plot, we found that most of the genes in cluster 2 show an upregulated compared to cluster 0. And based on these upregulated genes, we can find that these upregulated genes are lying on the response to viruses (*i.e.* Zika), influenza, and tumor cells (*i.e.* TNFAIP3). Especially, cluster 2 shows a strong level of Nkg2C which is upregulated in NK cells except for NK-cell precursors (NKP) [29]. The result is consistent with our hypothesis before: the NK cells from cluster 0 are short of feature expression which might be caused by artifact errors or their cell stage. To simplify the discussion, we name cells from cluster 0 as "outliers" since they have an abnormal distance from most of the NK cells.

Next, we run scMinerva on this dataset to observe GCN's functionality for the framework following the hyperparameter listed in Method Framework Hyper-parameters. Briefly, we generate walks for 20 times on each node with 42 as the walk length. And by comparing the classification result before and after training, we find that all the 30 cells from cluster 0 are correctly classified after training while only 6 of them are correct before training.

We visualized NK cells and Naive CD4 T cells from GSE156478 by RNA data, ADT data, and the trained embeddings from scMinerva as listed in Figure 4**c**. The first row is the complete graph while the second row is a zoom-in view only preserving the outliers to show their local positions. The reason we also visualized Naive CD4+ T cells is that, as shown in the zoom-in view of ADT's visualization, outliers are close to Naive CD4+ T cells since Naive CD4+ T cell is in an under-differential stage which is short of important surface proteins. The close spatial relation between the outliers and Naive CD4+ T cells reflects the low ADT expression of the outliers. But in the RNA plot, the outliers share an in-bound similarity with other NK cells and can be easily classified to the correct label. The visualization result after training also shows that scMinerva effectively classifies the outliers as NK cells as they are gathered with the main body of NK cells.

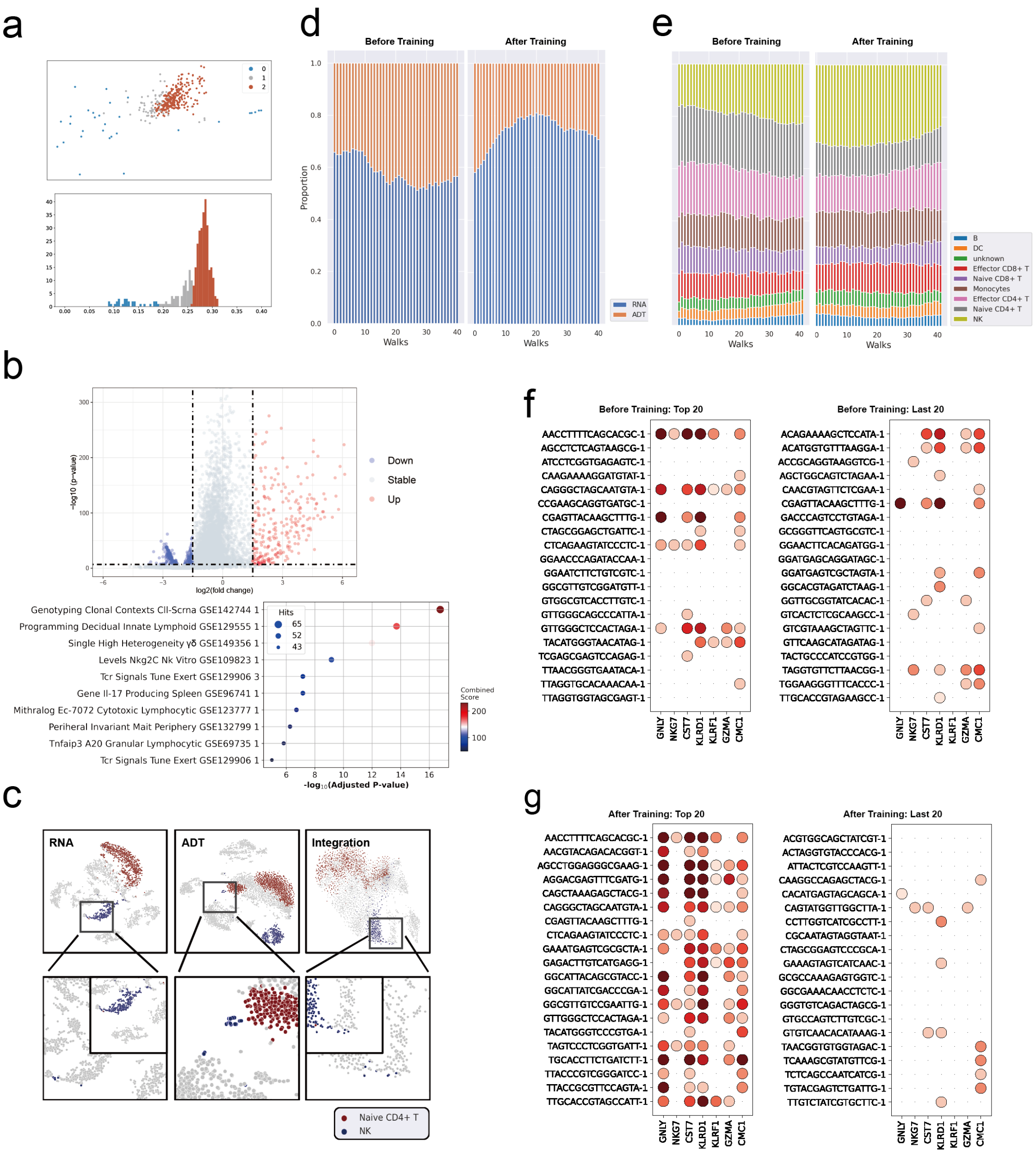

Figura 4: Model interpretability. **a**. We take 428 Natural Killer cells(NK cells) and cluster them by fitting their normalized library size with Gaussian Mixture Model. The scatter shows that most of the NK are in cluster 1 and 2 which have a reasonably high library size as shown in the histogram. Cells in cluster 0 have an extremely low library size. **b**. The gene expression level analysis between cluster 0 and 2. Cells from cluster 0 are short of important surface proteins on immune-system process. **c**. The visualization of NK cells on raw RNA data, raw ADT data, and scMinerva's embeddings. Cells from cluster 0 are plot in the second row after zooming-in. They are closer to CD4 Naive T cells in the plot of raw ADT which is misleading for classification **d**. The occurrence frequency for different omics on walks containing cluster 0 cells. After training, the random walk generates more walks on the RNA side which leads to a correct classification. **e**. Same as **d**, but is occurrence frequency for different cell types. After training, walks contain more NK cells and less CD4 Naive T cells which benefit the output embeddings. **f**. Nodes expression level on marker gene before training. There is no significant differences between the 20 cells with the most frequent occurrence in walks and the least 20. **g**. Same as **f** but for walks after training. The top 20 nodes show a strong upregulation on marker genes while the last 20 express weakly. This fact interpret scMinerva's robustness on sparse data.

To interpret why scMinerva works and how GCN benefits the framework, we will analyze GCN's function by evaluating the changes on generated walks during training. Since GCN is hard to be straightforwardly interpreted, in our framework's design, we implement random walk which is a transparent probabilistic model and word2vec [15] which is better studied under maximum likelihood estimation. Random walk and word2vec jointly build a possible "window" to the "black-box" of GCN, so that we can learn the effectiveness of GCN on this problem semi-transparently.

Before everything starts, we first need to briefly demonstrate the mechanism of the word2vec model. For each walk, we name the nodes it contains as its "steps", *i.e.*, a walk of length 42 has 42 steps. In our framework, we input the generated walks into the word2vec model and word2vec will output embeddings for nodes contained in the walks. Notably, word2vec will form embeddings to ensure the close neighbors in walks and nodes that occur in the same walk have a small distance in the embedding space under maximum likelihood estimation. Therefore, to obtain a more reasonable embedding of nodes for the later classification, the input walks are supposed to form a high-quality neighborhood for each step. In other words, we need nodes from the same cell type to form steps more frequently and consecutively in a walk, so that they can be assigned to similar embeddings by word2vec and classified to the same class by KNN.

In the case of NK cells, we have concluded that the graph of RNA side has a better neighborhood for the outliers, while on the ADT side, the outliers are closer to Naive CD4 T cells. To avoid the outliers from being classified as Naive CD4 T cells, we need the generated walks containing the outliers to have more nodes from the same cell type with them (*i.e.*, NK cells). Consequently, as visualized in Figure 4**c**, to include more NK cells in consecutive steps of walks containing them, we need these walks to generate more steps on the RNA side whose neighborhood is rich in NK cells instead of the ADT side.

To examine GCN with the above purpose, we extract walks containing the outliers before and after training with amounts 16251 and 17226, respectively, and analyze the changes in occurrence frequency of components at the omics level and cell type level.

Figure 4**d** shows the counting proportion of ADT and RNA on average for all the walks containing the outliers. The x-axis represents steps. And the y-axis represents the omics occurrence proportion in one step as we count the occurrence for different omics and normalize results to a sum of 1. From the most left bar of the plot, we can first observe that before training, there are about 65% walks containing these outliers starting from the RNA side while after training there are around 60% walks starting from the RNA side. However, before training, even though these walks have a better start under the randomness of random walk, they have a higher chance to "jump" to the ADT side with the initial transition probability. But, after training, the walks containing these outliers are more likely to learn and explore their neighbors on the RNA side as shown in the right bar chart. This trend strongly meets our analysis before as GCN will help to adjust the omics-transition probability and generate walks more on the "high-quality" omics side for these NK cells.

Similarly, Figure 4**e** shows the counting proportion concerning cell types on average for all the walks containing the outliers. The x-axis represents steps. And the y-axis represents the occurrence proportion of cell types in one step as we count the occurrence for different cell types and normalize results to a sum of 1. We are most concerned about the changes in NK cells and Naive CD4+ T cells. It can be observed that, before training, more walks are starting from the Naive CD4+ T cells containing these outliers and result in a high proportion of Naive CD4+ T cells in walks. But after training, the proportion of Naive CD4+ T cells is greatly reduced and NK cell becomes the most frequently appeared cell type in walks. As we mentioned in the mechanism of word2vec, with such a result, word2vec can more accurately produce similar embeddings to the outliers as other NK cells, so that they are correctly classified as "NK cells" after training.

This phenomenon probably comes from the loss function we select. Intuitively, it disperses different clusters detected by K-means and leads to a better local gathering on the whole graph. We set the cluster number k to be slightly larger than the actual clusters which will better separate nodes under a smaller cluster size. Therefore, the cluster containing those outliers is pushed away from the cluster containing the Naive CD4 T cells. And as a result, GCN adapts the transition probability which leads to a more comprehensive result after training.

The component proportion changes at single-cell level interpret the effectiveness of GCN on sparse data

In the last section, we evaluate the effectiveness of GCN by analyzing walks at the omics level and cell-type level. It looks like walks only take advantage of the nodes' neighbor relationship. But surprisingly, we will now further reveal the bond between single-cell occurrence frequency in walks and their biologic

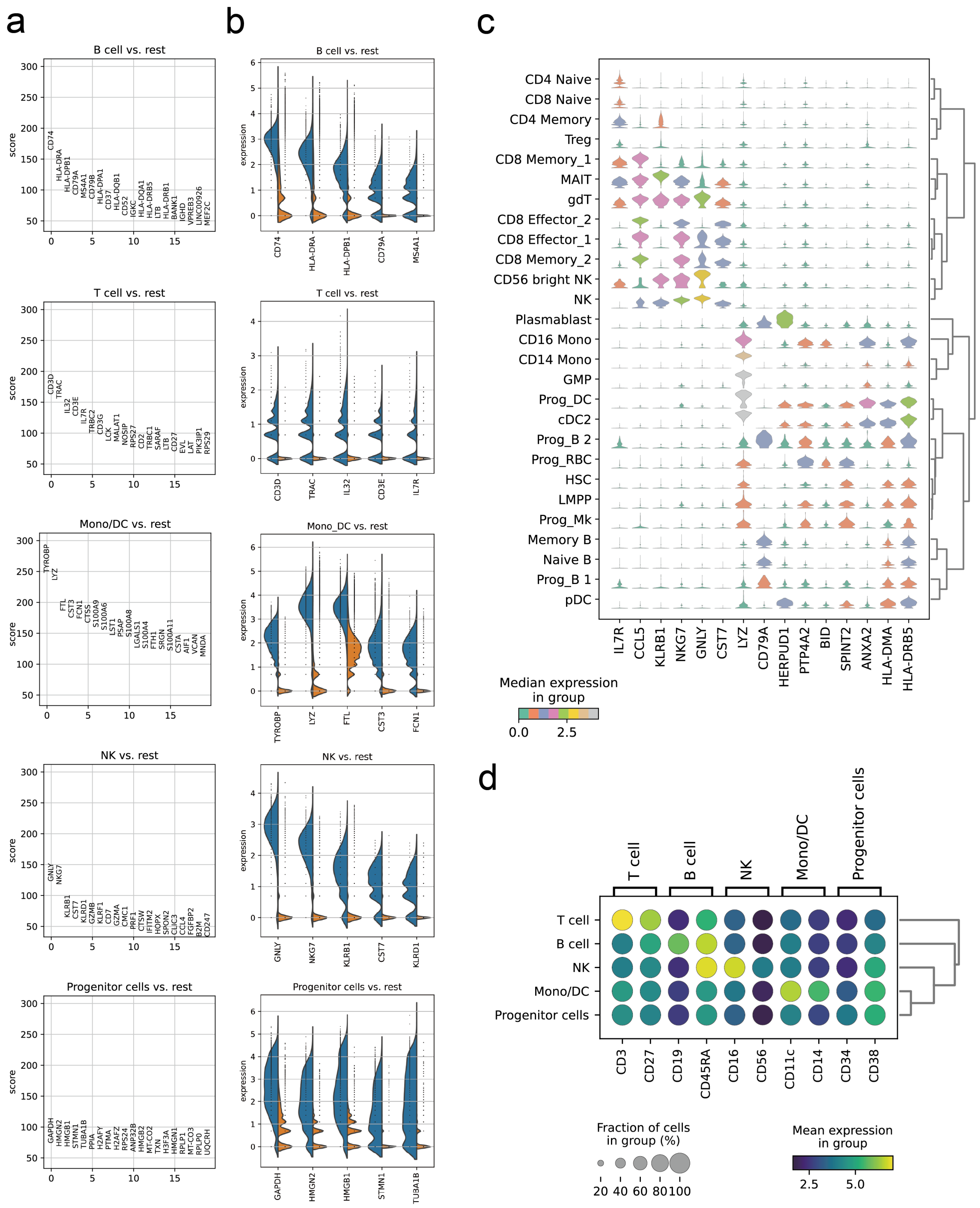

Figura 5: Biomarker detection on PBMC cells from GSE129639. **a**. The detected marker genes in 5 general types. **b**. The comparison of the expression level between the the top 5 detected biomarkers in the corresponding cell type and the rest of the cell types. **c**. Violin graph of the expression level of the biomarkers in fine-grained 27 subclasses. **d**. Dot graph showing the mean expression on ADT proteins expressed from some biomarkers in 5 cell types. The brackets in the upper of the graph indicate the detected high expression proteins for the corresponding cell types.

content expression level. This fact interprets the robustness of scMinerva on sparse data.

Our motivation is the following: if we consider the walks in a more fine-grained manner, each step of a walk is mapped to its cell type. Therefore, instead of only having the walks obtain steps, we can also obtain its mapping sequence of cell types each of which can be viewed as the state of the step. Now, we can compute the probability for a state to emit to some specific node. In another word, we group nodes by their cell types and count their occurrence frequency in walks for their cell type. We also run experiments on walks containing those outliers as discussed in the last section. To repeat, there are 16251 walks and 17226 walks before and after training, respectively.

Intuitively, 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 5. 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 4**f** is the expression level of the highest occurrence frequency of 20 nodes and the least 20 nodes before training. It is observed that their gene expression levels have no significant differences between the top 20 and the least 20. However, after training, as listed in Figure 4**g**, we found that the top 20 nodes are mostly under a high gene expression level while the last 20 cells are all under a low 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 as have a higher priority, GCN will assign a higher priority to high expression level nodes and more frequently utilize information from 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. It points out that the effectiveness of GCN has biological support with respect to nodes' expression levels. As we have mentioned in the last section, the GCN model helps random walk to learn from the high-quality omics according to different cells to achieve an anti-noise ability. In this section, our discovery further concludes that, under the sparse single-cell data, random walk also will assign a higher "priority", as known as the transition probability, to nodes with higher feature expression levels. These nodes are less influenced by the data sparsity or dropout.

Overall, scMinerva's process more relies on the nodes and omics which can most benefit its performance. With the strong interpretability, our method is practical for various downstream tasks. Here, we apply it to biomarker detection and cell-differentiation analysis to examine its practical value.

Predictions of scMinerva identify biomarkers of PBMC cells accurately

Biomarker acts as an indicator of biological processes and plays a vital role in disease detection [30]. With the predictions by scMinerva, we detect biomarkers on the predicted clusters by SCANPY [31]. 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 [32]. 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 [33, 34]. 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 [35], KLRB1 shows a suppression in human cancer tissues [36], and NKG7 regulates cytotoxic granule exocytosis and inflammation [37].

Finally, in Figure 5**d**, we use a dot plot to show the mean expression of the expressed protein for detected genes per cell type. The upper brackets on the graph indicate the high expression of ADT protein for the corresponding cell types. The brighter color indicates a higher expression level of the specific genes in the corresponding cell type, which aligns with the detection results shown in Figure 5**a** and 5b.

Predictions of scMinerva reveal potential differentiation changes after of naive immune cells infected COVID-19

In this section, with the predictions from scMinerva, we take human blood immune cells from dataset COVID-PBMC [22] to compare the differences in cell differentiation between cells infected with SARS-CoV-2 (COVID-19) and healthy cells. Our observations are also based on MELD [38]. By inputting the feature matrix of cells from the same cell type, we can infer their differentiation level in a single-cell resolution by MELD. Here, we are interested in the potential differentiation trend of Naive T cells for patients. The selected infected cells are from critical symptoms of human beings and can reflect the changes in the long run.

Generally, we first show that in this task, predictions of scMinerva greatly approximates the results obtained by ground-truth labels. Then, with our prediction, we will analyze the Naive T cells differentiation trend concerning different cell types. Finally, we will run gene enrichment analysis on cells from different cell types as well as on the same types of cells from healthy and infected tissues.

T cells are in an extremely important position for human immunity. However, most of the existing analyses on the influences for T cells caused by COVID-19 are on a cluster level. Researchers deduce the impact from the cell-type proportion changes observed in different symptom duration. But in our study here, we analyze the potential differentiation tendency of Naive CD4 T cells in a single-cell resolution by MELD and conclude results that strongly confirm some lately-proposed hypotheses at a single-cell level.

To avoid repeating, we only take $\mathrm{CD4}^+$ T cells in this section including $\mathrm{CD4}^+$ Naive, $\mathrm{CD4}^+$ $\mathrm{T_{CM}}$, $\mathrm{CD4}^+$ $\mathrm{IL-22}$, $\mathrm{CD4}^+$ prolif, $\mathrm{CD4}^+$ $\mathrm{T_H1}$, $\mathrm{CD4}^+$ $\mathrm{T_{EM}}$, $\mathrm{CD4}^+$ $\mathrm{T_H2}$, $\mathrm{CD4}^+$ $\mathrm{T_{FH}}$, MAIT and $\mathrm{CD4}^+$ $\mathrm{T_{reg}}$ to observe the potential cell differentiation changes after infection.

Firstly, we analyze the healthy cells. We input the raw expression data and the predicted labels from scMinerva into MELD and obtain the sample density for each cell concerning different cell types. The "sample density" is a kernel density estimate which estimates the likelihood of the sample label given the data. Its rows are cells and columns are cell types. Each entry of the sample density represents the kernel density estimate for a cell on some specific cell type. In our case, we input 10 cell types in the label uniquely, then the sample density will have ten columns concerning these types. Simply, the cell type that a cell obtains the highest sample density is the most likely type that the cell will differentiate to. Then we extract only $\mathrm{CD4}^+$ Naive cells and run Gaussian Mixture Model(GMM) with the number of components as 10 to sample density of these cells on $\mathrm{CD4}^+$ Naive column. The result is shown in Figure 6**a**. We can observe that, most of the cells have a high sample density on $\mathrm{CD4}^+$ Naive and only very few of them are below 0.05. In Figure 6**b**, we run the same procedure as before but only replace the input labels from ground-truth labels with predicted labels of scMinerva. It is obvious that the inference from our predictions greatly matches the annotations.

Followed this, in Figure 6**c**, we visualized the sample density score on some important functioning CD4 T cell types with predictions of scMinerva. For the 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 the Figure 6**c**. We provide a full graph concerning all ten classes in Appendix 8.

Then we run the same procedure on COVID-19 infected cells. Similarly, Figure 6**d,e** compare the GMM modeled sample density on $\mathrm{CD4}^+$ Naive column by ground-truth label and predictions of scMinerva respectively. Our prediction also shows a great approximation to the annotations. Followed 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

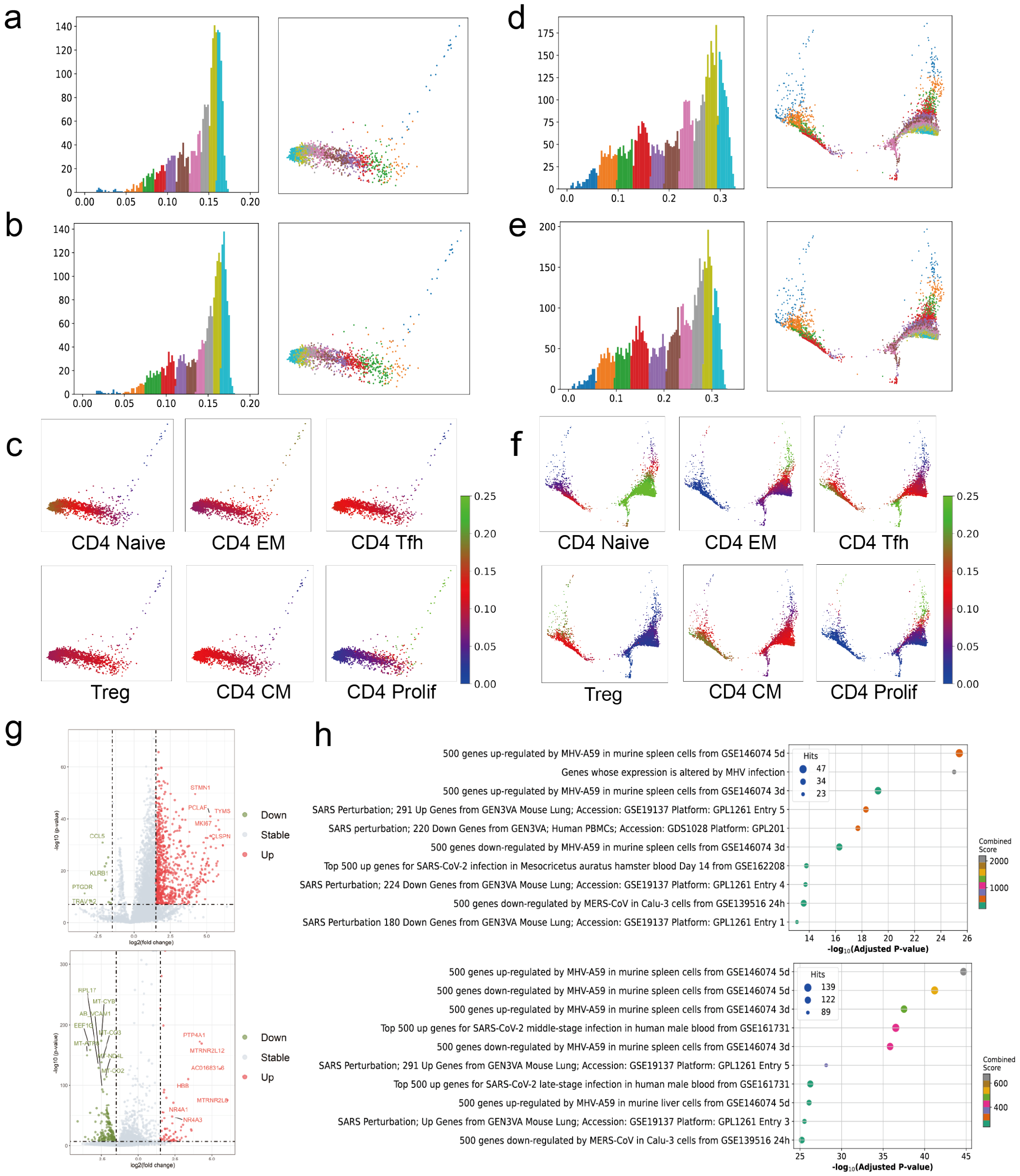

Figura 6: Cell differentiation analysis on CD4 Naive T cells on healthy samples and COVID-19 infected samples. **a**. The clusters fitted by Gaussian Mixture Model(GMM) on healthy tissues inferred by annotations. **b**. Same as **a**, but is inferred by predictions of scMinerva. Our method shows a strong 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$_{\mathrm{CM}}$, CD4$^+$ T$_{\mathrm{FH}}$, CD4$^+$ prolif are activated. **g**. Gene expression differences on CD4 Naive cells with different differentiation tendency between MAIT and CD4$^+$ T$_{\mathrm{EM}}$, and between CD4$^+$ Naive and CD4$^+$ T$_{\mathrm{reg}}$. We labeled some (down)upregulated genes in plots. **h**. Gene expression analysis between healthy cells and infected cells on CD4$^+$ $\mathrm{IL}-22$ and CD4$^+$ T$_{\mathrm{reg}}$. Infected cells perform a more active expression on Coronavirus-related genes.

cells especially to CD4$^+$ prolif, CD4$^+$ T$_{\mathrm{FH}}$, CD4$^+$ T$_{\mathrm{EM}}$, CD4$^+$ T$_{\mathrm{CM}}$ and CD4$^+$ T$_{\mathrm{reg}}$. We provide full graph with respect to all ten classes in Appendix 9.

The result is consistent with the observations from Jung J.H. *et.al.* [39]. They found a significant amount of cells differentiated into diverse memory subsets, comprising CD4$^+$ T$_{\mathrm{EM}}$ and CD4$^+$ T$_{\mathrm{CM}}$ compared to healthy tissue. Notably, their conclusions are from 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$_{\mathrm{H}}$1, CD4$^+$ T$_{\mathrm{H}}$2 and CD4$^+$ T$_{\mathrm{FH}}$. Cell proliferation is observed in various symptom duration. But it is more obvious for critical patients because of a T cell apoptosis [40].

In Figure 6**g**, we run gene expression difference analysis on CD4$^+$ Naive cells clustered by their maximum likelihood on differentiation tendency between MAIT and CD4$^+$ T$_{\mathrm{EM}}$, and between CD4$^+$ Naive and CD4$^+$ T$_{\mathrm{reg}}$. We observed some upregulated and downregulated genes as labeled in the plot. The difference is not as obvious as in mature cell types, however, on some significant marker genes, such as the detected marker gene CCL5 in Figure 5, there still have a large change fold among high genes expression cells. Bruth K.P. *et.al.* reported an approach to resolving unchecked inflammation and reducing SARS-CoV-2 plasma viral load via disruption of the CCL5-CCR5 axis [41]. These genes reflect the biological system changes and reveal significant potential for clinical research.

To observe the influences of COVID-19 infection more precisely, we analyzed the gap in gene expression levels between healthy tissue and infected tissue. The expression differentiation of CD4$^+$ $\mathrm{IL}-22$, and CD4$^+$ T$_{\mathrm{reg}}$ cells are listed as Figure 6**h**. Researchers have established a strong bond between coronavirus and murine hepatitis virus, such as A59 (MHV-A59). For the severe cases of COVID-19, it induces an extended inflammatory response that contributes to the increased morbidity [42]. A study on murine hepatitis virus helps to build knowledge of SARS-CoV-2 as well. From the results, it can be observed that the detected top differential genes significantly contribute to the immune process of MHV-A59 and other SARS viruses. Also, it shows that these upregulated genes are highly associated with a long COVID-19 symptom duration such as the middle-stage and late-stage.

Conclusion

In this study, we present scMinerva, an unsupervised single-cell multi-omics integration algorithm. It can flexibly handle any number of omics and is scalable to large datasets with efficient computational consumption. The experiments demonstrates its effectiveness on classification, especially the superiority on datasets with high noises and more omics.

We interpret the robustness of the model by analyzing the walks at omics level, cell-type level and sample level. The first two levels reveals the effectiveness of GCN on anti-noise while the analysis on the third level explains how GCN functions on sparse data. Also, to address its practical value, we performed biomarker detection of immune cells and analyze the changes in cell differentiation between healthy samples and samples infected with COVID-19 in a single-cell resolution.

Method

Problem Formulation

As different omics depict different aspects of a sample, scMinerva aims to integrate sample information from different omics together so that different biological counting can complement each other by being aware of the neighborhood. The whole framework is shown in Figure 1,

Naturally, the underlying relationship between different omics can be modeled by a graph structure and processed by graph convolution networks (GCN) [43]. Hence, we treat the embedding generation process as a multi-modal unsupervised graph merging and graph representation learning problem. Assume a dataset with c omics and n samples. Denote the samples set $S = \{s_i\}, i \in [1, n]$, where n is the number of samples. In our setting, for each omics, there is firstly a separate graph $\mathcal{O}_j,\ j \in [1, c]$ constructed based on K-Nearest Neighbor (KNN) distance of the coordinates of each spot, where the spots are regarded as nodes whose attributes are gene expression, protein counting, etc. (if applicable).

For $\mathcal{O}_j \in \{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_c\}$, $\mathcal{O}_j = (\mathcal{V}_j, \mathcal{E}_j)$ where $\mathcal{V}_j$ refers to its node set and $\mathcal{E}_j \subseteq (\mathcal{V}_i \times \mathcal{V}_j, \mathbf{R})$ is the set of weighted edges. The feature matrix of all the nodes is denoted as $\mathbf{X}_j = \{x_{1j}, x_{2j}, ..., x_{nj}\} \subseteq \mathbb{R}^{n \times F_j}$, where F_j is the dimension of node features.

Denote $\mathbf{o}(x)$ to be the index of omics which node x belongs to and s_{rj} is the mapping node of sample s_r in the jth omics. Therefore $\mathbf{o}(s_{rj}) = j$ for a sample s_j.

With $\{\mathcal{O}_j | j \in [1, c]\}$, we build a directed heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the node set and $\mathcal{E} \subseteq (\mathcal{V} \times \mathcal{V}, \mathbf{R})$ is the edge set. For node $u, v \in \mathcal{G}$, the weight from node u to node v is denoted as w_{uv} while the opposite is w_{vu}. $\mathcal{G}$ contains all the nodes from $\{\mathcal{O}_j | j \in [1, c]\}$ and therefore its node set size $|\mathcal{V}| = nc$. The feature matrix of all the nodes is denoted as $\mathbf{X} = \{x_1, x_2, ..., x_{kn}\} \subseteq \mathbb{R}^{kn \times F}$, where F is the dimension of node features.

For Graph Convolutional Networks (GCNs), let $\hat{\mathbf{A}}$ denote the normalized adjacency matrix and $\mathbf{H}^{(l-1)}$ denote the embedding of layer $l - 1$. The propagation of the GCNs is defined as $\mathbf{H}^{(l)} = \sigma(\hat{\mathbf{A}}\mathbf{H}^{(l-1)}\mathbf{W}^l)$, where $\mathbf{W}^l$ is a learnable weight matrix and σ is a non-linear activation function.

The training can be divided into three stages: heterogeneous graph construction, random walk and the training stage. In heterogeneous graph construction phase, separately constructed graph and an omics-to-omics transition weight matrix are used for constructing the heterogeneous graph $\mathcal{G}$. $\mathcal{G}$ is a weighted directed graph constructed from $\{\mathcal{O}_j | j \in [1, c]\}$ that contains a global graph topology of all the omics. In the random walk stage, the heterogeneous graph $\mathcal{G}$ is input. We run random walk on the graph with a defined transition probability P and obtain the embeddings by inputting walk matrix $\mathbf{J}$ into the word2vec model [15]. In the training stage, the embedding $\mathbf{X}$ produced from last stage and the graph adjacency matrix of $\{\mathcal{O}_i\}$ is input into the $GCN(\cdot)$ model. The model outputs the optimized omics-to-omics transition matrix $\mathbf{T}' \subseteq \mathbb{R}^{nc} \times c(c + 1)$ to continue the iteration.

Heterogeneous Graph Construction

With the omics specific graphs $\{\mathcal{O}_j | j \in [1, c]\}$ and a omics-to-omics transition matrix $\mathbf{T}$ as inputs, we build a heterogeneous graph $\mathcal{G}$. For all s_{rt} and s_{rl}, where $r \in [1, n]$ and $t \neq l$, assign

$$w_{s_{rv} s_{rl}} = \mathbf{T}[r][j \cdot k + l].$$

In the initialization of the first epoch, we by default set $\mathbf{T} = \mathbf{a}^{n \times k(k+1)}$ where $\mathbf{a}$ is a constant. Therefore, $\mathbf{T}$ is a matrix filled by a constant value $\mathbf{a}$ of shape $(n, k^2 - k)$. In another word, we first link all the nodes from the same sample in different omics together and assign an initial weight $\mathbf{a}$ to these edges. the new graph satisfies

$$|\mathcal{V}| = \sum_{j=1}^{k} |\mathcal{O}_j| \text{ and } |\mathcal{E}| = \sum_{1}^{k} |\mathcal{E}_i| + nc(c + 1).$$

To emphasize, $\mathcal{G}$ is a directed, weighted graph. the initialized constant weights will be adjusted to an optimal stage by the GCN feature which is demonstrated later.

Random Walk

In random walk, we define a transition probability function $P(u|v)$ represents the transition probability from source node u to the next node v. We generate m walks of length l started from each graph node. In total, there are ncm walks generated and each has length l. It can be represented as a walk matrix $\mathbf{J} \subseteq \mathbb{R}^{ncm} \times \mathbb{R}^{l}$.

On network $\mathcal{G}$, we simulate random walks of a fixed length l on given source node u. Denote w_i as the ith node in the walk starting from $w_0 = u$. For some $g \in [1, l]$, nodes $\{w_g\}$ are generated by

$$P(w_g = x | w_{g-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z} & \text{if}(v, x) \in \mathcal{E} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where π_{vx} is the unnormalized transition probability from node v to node x, and Z is a normalizing constant.

The normalizing constant Z need to be carefully selected which can guide the search to explore different types of neighborhoods. Instead of node2vec which exhibited a mixture of Breadth-First Search(BFS) and Depth-First-Search(DFS), we define a 2^{nd} order random walk with three parameters p, q and z which guide the walk: Consider a random walk that just traveled from node t to node v. The walk now are determining the next step so it evaluates the transition probability π_{vx} on edge (v, x) leading from v. We set the unnormalized transition probability as $\pi_{vx} = \alpha_{pqz} \cdot w_{vs}$, where

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

and d_{tx} represents the shortest distance between node t and node x.

To recall, m is the amount of walks generated from each graph node. So when the random walk finishes, these are $|\mathcal{V}| \cdot m$ walks in total. Since each walk is of length l, the walk matrix $\mathbf{J} \subseteq (|\mathcal{V}| \cdot m) \times l$ and is the input of the word2vec model. Let $f : S \to \mathbb{R}^F$ be the mapping function from sample sets to embeddings of dimension F. We aim to learn such representations for the downstream prediction task. For source node $u \in \mathcal{V}$, define $N_{sample}(u) \subset \mathcal{V}$ are the neighborhood of node u generated through a sample of walks. With a skip-graph architecture, we are trying to find the f gives

$$\max \sum_{u \in \mathcal{V}} \log Pr(N_{sample}(u) | f(u)). \tag{3}$$

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

$$\max_f \sum_{u \in \mathcal{V}} [-\log K_u + \sum_{n_i \in N_{sample}(n)} f(n_i) \cdot f(u)]. \tag{4}$$

where $K_u = \sum_{v \in \mathcal{V}} \exp(f(u) \cdot f(v))$ is the partition function for nodes. Solve Eq. 4 using stochastic gradient ascent (SGD) over the model defining the features f. The output feature representations from f is denoted as $\mathbb{M} \in \mathbb{R}^{|\mathcal{V}| \times F}$.

Model Training

A GCN of β layers is built here by stacking multiple convolutional layers. Each layer is defined as

$$\mathbf{L}^{l+1} = g(\mathbf{L}^{(l)}, \mathbf{A}) = \sigma^{(l)}(\mathbf{A}\mathbf{L}^{l}\mathbf{W}^{(l)}), \tag{5}$$

where $\mathbf{L}^{(l)}$ is the input of the lth layer and $\mathbf{W}^{(l)}$ is the weight matrix of the lth layer. $\sigma(\cdot)$ denotes an activation function which is non-linear. Since the output is edge weight, $\sigma(\cdot)^{(\beta)}$ is a non-negative activation function.

The adjacency matrix $\hat{\mathbf{A}}$ is rewrote by Kipf and Welling [43] as

$$\hat{\mathbf{A}} = \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} = \hat{\mathbf{D}}^{-\frac{1}{2}} (\hat{\mathbf{A}} + \mathbf{I}) \hat{\mathbf{D}}^{-\frac{1}{2}}. \tag{6}$$

$\hat{\mathbf{D}}^{-\frac{1}{2}}$ is the diagonal node degree matrix of $\hat{\mathbf{A}}$ and $\mathbf{I}$ is the identity matrix. We build adjacency matrix from the edge list $\mathcal{E}$ and input it to GCN model with embedding $\mathbf{M}$. Here, $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ and $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times F}$. The output edge weight matrix can be written as:

$$\mathbf{T}' = GCN(\mathbf{M}, \hat{\mathbf{A}}). \tag{7}$$

To guide the $GCN(\cdot)$, we adopt the loss function from DeepCluster [16] which supervises the model by producing pseudo-labels. For the output embeddings $\mathbf{M}$, we run k-means to take a set of vectors as input and cluster them into k different groups based on the geometric neighborhood. Therefore, each node s_{ij} is associated with a label l_{ij} in $\{0, 1\}^k$. k-means jointly formulate a $F \times k$ centroid matrix $\boldsymbol{\nu}$ and cluster nodes by solving:

$$\min_{\boldsymbol{\nu} \in \mathbb{R}^{d \times k}} \frac{1}{|\mathcal{V}|} \sum_{i=1}^{|\mathcal{V}|} \| f_\theta(x_n) - \boldsymbol{\nu} y_n \|^2 \quad \text{such that} \quad y_n^T \mathbf{1}_k = 1. \tag{8}$$

Here, $\mathbf{1}_k$ represents a all one vector of length k. Solving Eq. 8 provides a set of optimal pseudo-labels (l^*_{ij}). First, a Student's t-distribution kernel is used to calculate the soft assignment probability q_{ij} of the embedding $\mathbf{M}_i$ to the cluster centroid $\boldsymbol{\nu}_i$

$$q_{ij} = \frac{(1 + ||\mathbf{M}_i - \boldsymbol{\nu}_i||^2)^{-1}}{\sum_{j'}(1 + ||\mathbf{M}_i - \boldsymbol{\nu}_{j'}||^2)^{-1})}. \tag{9}$$

Next, based on q_{ij}, a target distribution P is calculated to help learn from the assignments with higher scores

$$p_{ij} = \frac{q_{ij}^2/\sum_i q_{ij}}{\sum_j^{'}(q_{ij'}^2/)\sum_i q_{ij'}}. \tag{10}$$

Finally, the loss function is defined as

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}. \tag{11}$$

After training, the output $T' \in \mathbb{R}^{|\mathcal{V}| \times c(c+1)}$.

After training, with the walks generated on $\mathcal{G}$, we map the nodes in walks back to its sample index.

$$\hat{\mathbf{w}}_i = s_j \text{ if } \mathbf{w}_i = s_{jr} \tag{12}$$

where $j \in [1, n]$ and $i \in [1, k]$. Input the remapped $\hat{\mathbf{W}}$ to word2vec model. Let $f : S \to \mathbb{R}^F$ be the mapping function from sample sets to embeddings of dimension F. With a skip-graph architecture, we are trying to find the f gives

$$\max \sum_{u \in S} \log Pr(N_{sample}(u)|f(u)). \tag{13}$$

With the optimal f, the output embedding is $f(S) \in \mathbb{R}^{|S| \times F}$.

Framework Hyper-parameters

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 $p = 1$, $q = 2$ and $z = 0.7$. Also, 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 equals to δ before normalizing. We set δ to 0.1.

After setting up the transition probability, on each node of the graph, we run random walk start from it for 20 times. Each time it will generate a walk of length 42. With the generated walks, we input them to word2vec under algorithm CBOW and window size 10. Word2vec will output embeddings of dimension 64 for nodes contained in walks.

GCN Model

We input the embeddings as well as the edge list of the heterogeneous graph into the GCN model and GCN will output the weight matrix of omics-transition links. The GCN in scMinerva consists of two fully connected layers. The first layer has 64 nodes, while the second layer has 16 nodes. The *ReLU* function, defined as $ReLU(x) = \max(0, x)$, is used as the nonlinear activation function after the linear transformation.

The GCN is led by a deep clustering loss function. Suppose the input dataset has c omics and t cell types, we run k-means with the number of class equal to $t \cdot (c+1)$ and the number of time the k-means algorithm will be run with different centroid seeds equal to 5. Then we calculate target distribution by Equation 10 and calculate the loss value by Equation 11. Both of the calculations have a smooth value 1e-6 on the denominator to avoid the zero division. We update the target distribution every 4 epochs. For the optimizer, we use SGD optimizer with the learning rate as 1e-4 and the momentum coefficient m is 0.1. We train the GCN for 20 epochs.

The hyper-parameters are determined using grid search with cross-validation.

Classification

We conduct experiments to evaluate the classification of scMinerva and other methods on simulated data and real-world data, and scMinerva outperforms others in each set. For two-omics data, we can see that scMinerva has the best performance on accuracy (ACC), F1-macro, and ARI (adjusted rand index) (Figure 2**a**. Intuitively, the clusters generated by scMinerva are clearer and have smoother edges (Figure 2**b**). We also demonstrate these methods on four three-omics datasets with six sets of annotations. Firstly, we implement each method to obtain the embedding of the corresponding dataset, and we split each one into training sets and testing sets. The training set is fed to a KNN classifier, and we validate the model on the corresponding test set. We evaluate four approaches only using a training set 95%, 90%, and 80% respectively. scMinerva outperforms other methods on ACC (Figure 2**d** and appendix Figure 7). It demonstrates that scMinerva can integrate multi-omics effectively with a extremely low requirement on training labels. To further demonstrate the robustness of scMinerva, we simulate several four-omics datasets and scMinerva still has a better performance compared with other methods (Table 1).

We use accuray, weighted F1-score, macro F1-score and adjusted rand index to fully evaluate the classification performance of methods, details of the metrics can be found in appendix section Evaluation metrics.

Data Simulation

We generate a four-omics dataset based on the synthetic RNA-seq generated by Splatter [19]. 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 [44]. 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 number 2k, 5k, 10k, and 30k, respectively. The simulated RNA, ATAC, ADT from RNA and ADT from ATAC data are of feature number 815, 2613, 227 and 227 respectively.

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.

Availability of data

The datasets analyzed in this study are available under the following accession numbers: GSE128639 [21], GSE156478-CITE [20], GSE156478-ASAP [20]), COVID-PBMC [22]), SNARE-seq [23] and scNMT-seq [24].

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no competing interests.

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.

Referências

[1] Keren-Shaul, H., Spinrad, A., Weiner, A., Matcovitch-Natan, O., Dvir-Szternfeld, R., Ulland, T.K., David, E., Baruch, K., Lara-Astaiso, D., Toth, B., *et al.*: A unique microglia type associated with restricting development of alzheimer's disease. Cell **169**(7), 1276–1290 (2017)

[2] Bäumer, C., Fisch, E., Wedler, H., Reinecke, F., Korfhage, C.: Exploring dna quality of single cells for genome analysis with simultaneous whole-genome amplification. Scientific Reports **8**(1), 1–10 (2018)

[3] 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)

[4] 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)

[5] Tewari, A., Zollhofer, M., Kim, H., Garrido, P., Bernard, F., Perez, P., Theobalt, C.: Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1274–1283 (2017)

[6] 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)

[7] 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)

[8] 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)

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

[10] 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)

[11] 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)

[12] 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)

[13] 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)

[14] 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)

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

[16] 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)

[17] Liu, S., Park, J.H., Yoo, S.: Efficient and effective graph convolution networks. In: Proceedings of the 2020 SIAM International Conference on Data Mining, pp. 388–396 (2020). SIAM

[18] 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)

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

[20] 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)

[21] 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)

[22] 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)

[23] 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)

[24] 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)

[25] 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)

[26] Stoeckius, M., Zheng, S., Houck-Loomis, B., Hao, S., Yeung, B.Z., Mauck, W.M., Smibert, P., Satija, R.: Cell hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. Genome biology **19**(1), 1–12 (2018)

[27] Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)

[28] Huntington, N.D., Vosshenrich, C.A., Di Santo, J.P.: Developmental pathways that generate natural-killer-cell diversity in mice and humans. Nature Reviews Immunology **7**(9), 703–714 (2007)

[29] Abel, A.M., Yang, C., Thakar, M.S., Malarkannan, S.: Natural killer cells: development, maturation, and clinical utilization. Frontiers in immunology **9**, 1869 (2018)

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

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

[32] 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)

[33] 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)

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

[35] 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)

[36] 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)

[37] 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)

[38] 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)

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

[40] 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)

[41] Patterson, B.K., Seethamraju, H., Dhody, K., Corley, M.J., Kazempour, K., Lalezari, J., Pang, A.P., Sugai, C., Francisco, E.B., Pise, A., et al.: Disruption of the ccl5/rantes-ccr5 pathway restores immune homeostasis and reduces plasma viral load in critical covid-19. MedRxiv (2020)

[42] 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)

[43] Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

[44] 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)

[45] 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)

[46] 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)

[47] 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)

[48] 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)

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

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

Appendix

Preprocessing

GSE128639

The expression matrices was used as quantified in the original experiment [45]. For gene expression, standard log-normalization with default parameters in Seurat [21] 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 [13].

scNMT

Gene counts were quantified from the mapped reads by featureCounts [46], and gene annotations were pbtained from Ensembl version 87 [47]. 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 [24].

SNARE

SNAREseq [23] 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 [48]. 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 [48].

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) [49] as

$$ACC = \frac{TP + TN}{P + N}. \tag{14}$$

F1 score Here, F1 score can be calculated [50] as:

$$F1\ score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}. \tag{15}$$

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

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max[\mathrm{RI}] - \mathbb{E}[\mathrm{RI}]} \tag{16}$$

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

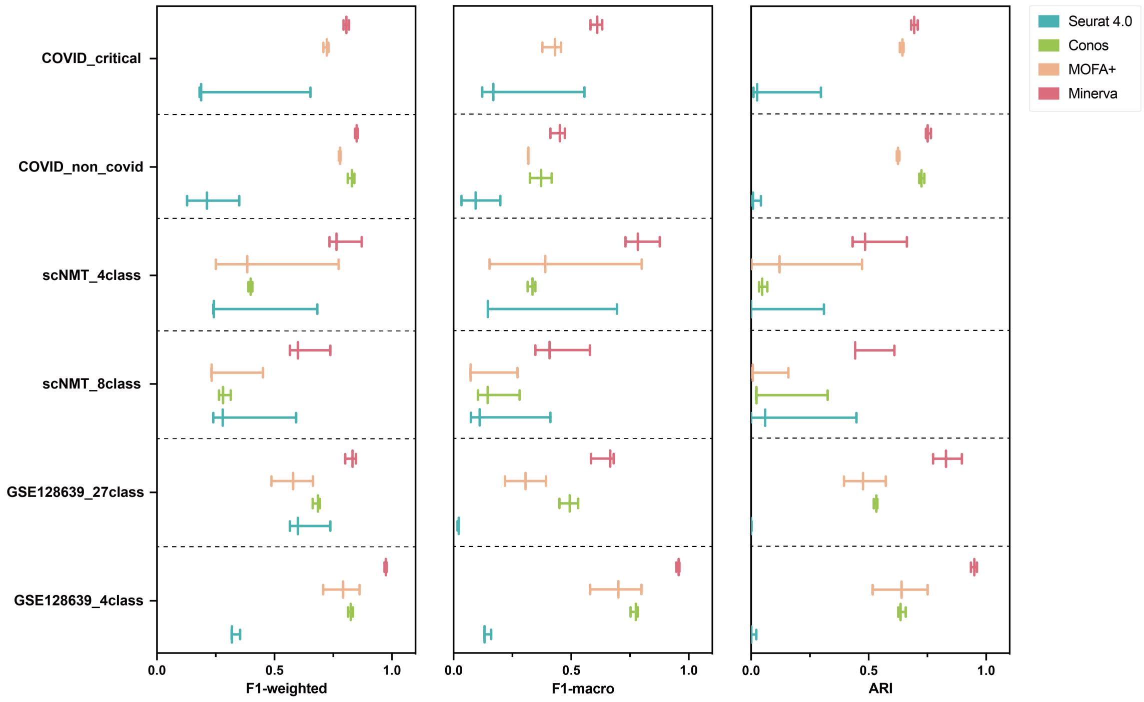

Figura 7: Comparison of classification F1-weighted score, F1-macro score and ARI on four datasets with six set of annotations. Each row contains three ticks which represent a method's performance on a dataset with test size 95%, 90% and 80% respectively.

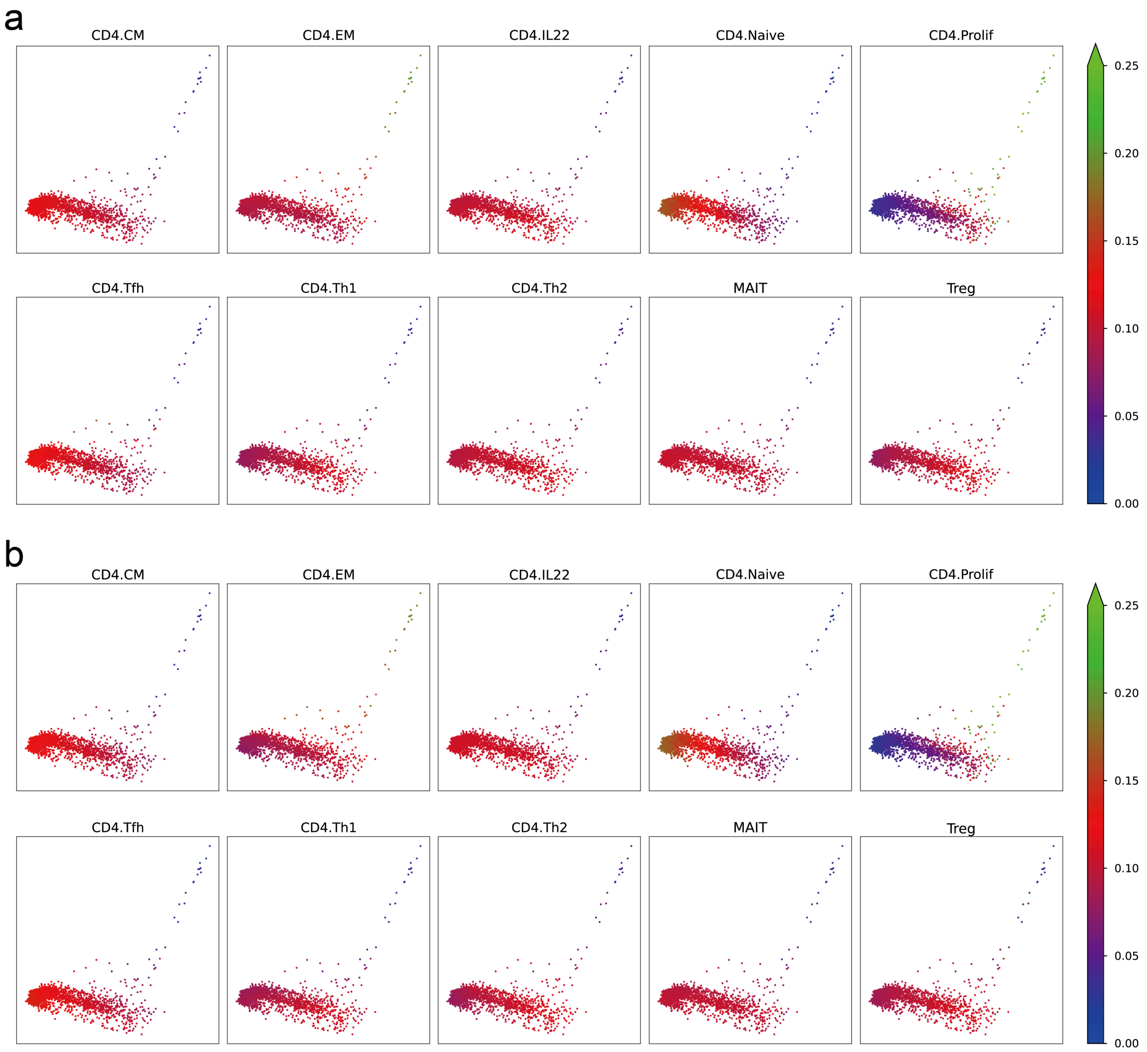

Figura 8: The cell differentiation tendency analysis on healthy cells. **a**. The differentiation score on CD4 Naive cells to different cell types inferred from ground-truth label. **b**. Same as **a** but is interred from scMinerva's predicted label. In all the cell types, our method shows a strong approximation to the annotation's result.

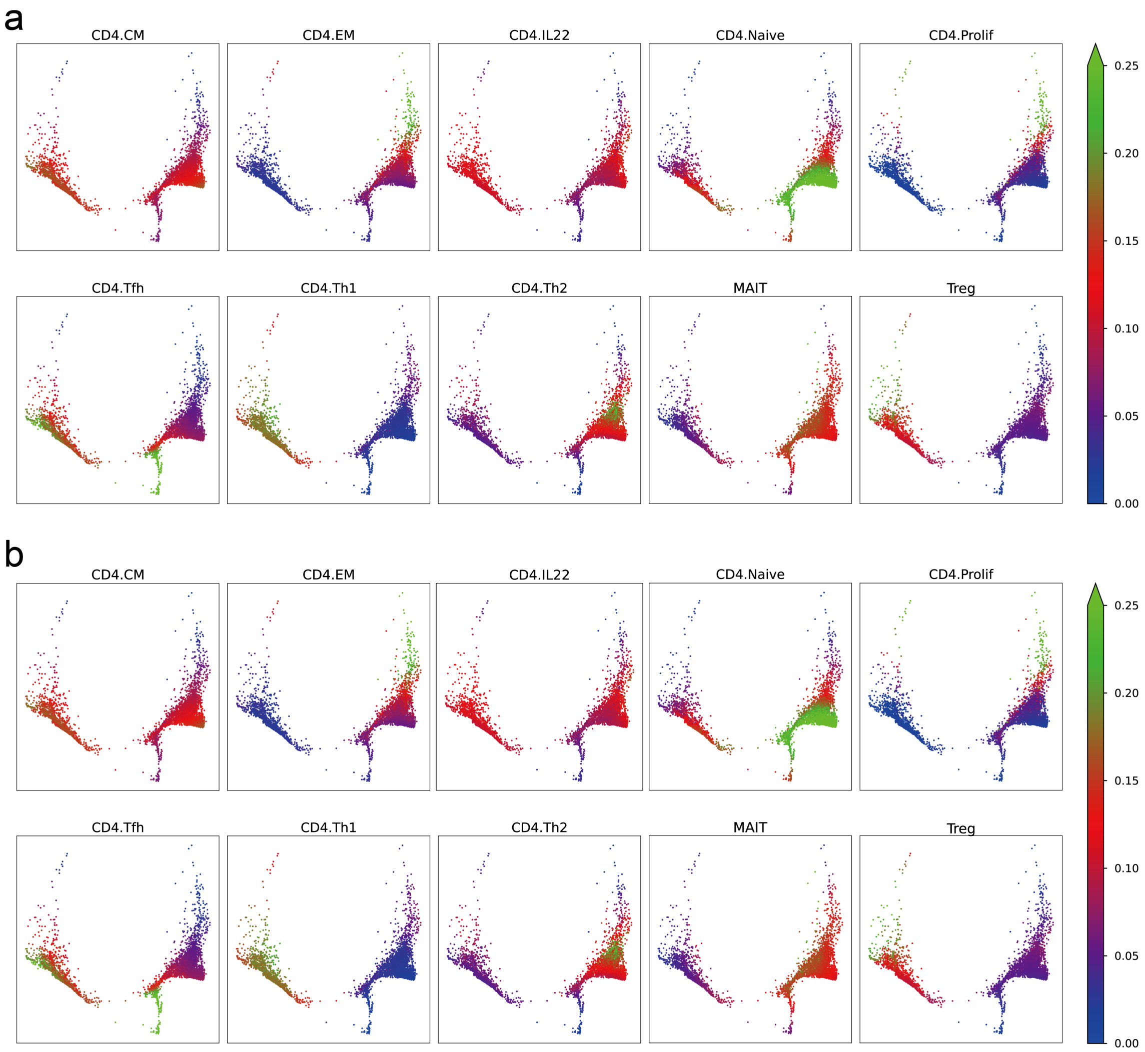

Figura 9: The cell differentiation tendency analysis on infected cells. **a**. The differentiation score on CD4 Naive cells to different cell types inferred from ground-truth label. **b**. Same as **a** but is interred from scMinerva's predicted label. In all the cell types, our method shows a strong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