Optimization and redevelopment of single-cell data analysis workflow based on deep generative models

Author
Yunhe Liu1,†, Qiqing Fu1,†, Chenyu Dong1, Xiaqiong Xia1, Gang Liu1,*, Lei Liu1,2,*
1 Institute of Biomedical Sciences, Fudan University, Shanghai, P.R. China
2 School of Basic Medical Science, Fudan University, Shanghai, P.R. China
† First Author
* Correspondence Author: liugang@fudan.edu.cn, liulei_sibs@163.com

Abstract
The present single-cell RNA sequencing (scRNA-seq) analysis pipelines require a combination of appropriate normalization, dimension reduction, clustering, and specific-gene analysis algorithms, but the rationale for the choice of these algorithms is relatively subjective because of the lack of ground truth assessment conclusions. As the number of captured single-cells increases, the number of different types of noise cells also increases, which can strongly affect the analysis efficiency. For scRNA-seq, a technology that generates data through multi-process operations, the deep generative model should be a good choice for this type of data analysis, allowing simultaneous estimation of multiple unobservable parameters assumed in the data generation process. Hence, in our study, we sequenced a pool of pre-labeled single cells to obtain a batch of scRNA-seq data with main and fine labels, which was then used to evaluate the clustering and specific-gene analysis methods. Afterward, we applied two deep generative models to infer the probabilities of pseudo and impurity cells. And by stepwise removing the inferred noise cells, the clustering performance and the consistency of different specific-gene analysis methods are both greatly improved. After that, we applied Deep-LDA (a latent Dirichlet allocation-based deep generative model) to scRNA-seq data analysis. And this model takes the count matrix as input, and makes the classification and specific gene optimization process mutually dependent, which has more practical sense and simplifies the analysis workflow. At last, we successfully implemented the model with transferred knowledge to make single-cell annotation and verified its superior performance.

Introduction
ScRNA-seq has grown to be the primary approach for research on histo-differentiation1, brain structure2, and tumor immunology3,4,5,6. However, in practice, the outcomes generated from this data type are frequently unstable7. The major reason is that the obtained data which was sequenced from a huge number of single cells, is insufficient to cover the whole transcriptome feature of each cell, leading to a high 0-inflation ratio and technical noise8. Moreover, the transcripts of different cell types have poor overlap, causing a more complicated distribution in observed data.

Many algorithms have been developed to reduce the noise within and between the cells. BASiCS9 and Scran10 estimate a specific scale factor for normalization. MNNA11 has been employed to integrate data from multiple batches. DCA12, scVI13, and ZINB-WaVE14 infer noise-free expression values with generative models. And SCTTransform15 corrects the dependence between gene expression values and sequencing depths by means of regularized negative binomial distribution regression. The gains from these methods are remarkable for the current clustering and specific gene analysis. But for these methods reshaping observed expression value, there is a high possibility of eliminating some useful information when increasing the positive signal in their assumption.

The commonly used single-cell clustering procedures work on the cell nearest neighbor network by KNN and SNN, and then utilize the community-detection algorithms16, such as louvain17 and leiden18, to iteratively optimize
cell clusters. And the existed noise cells which distributed widely separated from the target cells in latent space, such as pseudo cells that capture environment RNA\textsuperscript{19} and bits of impurity cells, can largely affect cell neighbor relationship estimation and clustering iteration. Thus, removing these noise cells will theoretically improve the performance of the current clustering algorithms. The specific gene analysis depends on clustering analysis, which in turn is not. That is to say, the quality of clustering will directly affect the specific genes' quality, while the generated class-specific genes cannot reversely optimize the clustering. However, in the practical sense, the cell-specific genes should be the law for their sorting. So, it will be more meaningful to enable the two analysis steps to be coupled and mutually dependent.

The scRNA-seq data analysis pipelines consist of several steps\textsuperscript{20}, each of which has many optional algorithms to choose from\textsuperscript{21}. Due to lack of suitable ground truth, it’s hard to objectively evaluate these pipelines\textsuperscript{16, 22}. Although the simulation of scRNA-seq data drawing from Splatter\textsuperscript{8}, SPARSim\textsuperscript{23}, SymSim\textsuperscript{24}, etc., can produce the pseudo-ground truth for algorithm evaluation, they only reproduce the limited assumed features of the actual data. Instead, the real pre-labeled pooled sample scRNA-seq data will be a more optimal choice of ground truth.

Deep generative models\textsuperscript{13} can combine assumed multiple probability distributions and deterministic relations to generate observed data, and infer the unobservable parameters in the generation process by the variation inference or MCMC\textsuperscript{13}. Various types of controlling parameters, such as scale factor, real RNA signal, noise cell probability, etc., can be incorporated into the model generation process and inferred simultaneously. Such models have been shown to be highly suitable for fitting the scRNA-seq data which have complicated features\textsuperscript{25, 26, 27}. And some modules of generation are constructed based on the neural network architecture, which have a higher ability to fit the nonlinear elements in data\textsuperscript{28, 29}.

Regarding the above issues, we experimentally obtained a batch of scRNA-seq data with main and fine labels and used it as ground truth to evaluate the analysis pipelines. And then, we used deep generative models for the inference of two types of noise cells and used an LDA-based deep generative model\textsuperscript{30} for cell classification and class-specific gene analysis. And finally, we validated the models’ superior performance and developed more related applications.

A batch of sequencing data from pre-enriched labeled single-cell

Following the procedures of pre-enrichment, labeling, and then pooled sequencing, we obtained a batch of scRNA-seq data from mixed 10 types of immune cells and a batch of bulk-seq data from 8 of them (Fig 1a; Methods). After preprocessing, dimension reduction and visualization of the two types of data (Methods), the distribution of the cell types in scRNA data (Fig 1b) is spatially consistent with that in bulk-seq data (Fig 1d left). Subsequently, we employed CIBERSORT\textsuperscript{31} to decompose the bulk-seq data (Fig 1d right, Methods), and also verified the consistency. After that, we collected 460 specific genes (Methods; Table S8) from 3 databases and 4 references for the 10 types of immune cells. And the differential analysis verified that most of the collected genes were significant in both types of sequencing data, except for some sub-categories of T cells (Fig 1c; Table S8). The heatmap of robust specific genes in the two types of data (Fig 1b; Fig S1a) and the UMAP of some key gene intensities (Fig S1b) showed that the genes have high specificity in corresponding cell types. In summary, we suggested that the batch of scRNA-seq data with main and fine labels can be used as ground truth for classification algorithm evaluation.
Figure 1 Analysis of pre-labeled immune cell sequencing data

a. Schematic diagram of the sequencing process of pre-enriched labeled immune cells; b. UMAP visualization of the pooled scRNA-seq data, color-coded labels for pre-enriched cell types; c. Bar plot of the specific genes in prior knowledge (grey) and two small bars showing which fraction is significant after differential analysis (Table S8) in single-cell data (red) and bulk data (green), respectively (the horizontal overlap between the three bars is the intersection domain). Independence tests (fisher test) were applied to examine the relationship between the differential analysis results and the prior knowledge, and its significance was marked above or below the corresponding bar (`***`, `**`, `*`, `∙`, `NS` represent P values <0.001, <0.01, <0.05, <0.1, >0.1, respectively); d. UMAP visualization of bulk sequencing data (left panel) and estimated cell-type component proportion (right panel). The color panel for cell types is the same as a; e. Robust specific-gene heatmap for bulk sequencing data.

Clustering methods comparison and improvement after inferred noise cell removal

After removing the inferred pseudo-cells and impurity cells by two deep generative models, the remaining data were visualized by UMAP and tSNE (Fig 2a; Fig S2a, b; Methods). The cell spatial distribution showed that the mixed-type cell distributed area (Fig 1b) was inferred as pseudo-cells (Fig 2a left; Fig S2b left), which were composed of the same ambient RNA under different cell type labels, and some small patches of cells sporadically distributed in the plot (Fig 2a middle; Fig S2b middle) were inferred as impurity cells. And with stepwise removal of the noise cells, the aggregation degrees for different types of cells were enhanced, especially for Memory CD8+ T cells (Fig 2a). Then, we applied the 3-phase data to evaluate the effectiveness of 5 single-cell clustering procedures with 13 clustering metrics (Methods). And the evaluation results indicated that the noise cell removal significantly improved
the clustering scores for all procedures (Fig 2b; Fig S3a-f). And the overall scores of louvain-, leiden- and densitypeak34-based clustering were outstanding and comparable, which we then suggested as the first echelon methods. While DDRTree35- and Consensus36-based clustering were seriously affected by noise and the overall scores were lower. In addition, with noise cell removal, the consistency between the distribution of annotated cluster and corresponding real cell type was improved on main cell labels (Fig 2c left, Fig S4), while poorly improved on fine cell labels which present more irregular shape within the main label distributed area, such as Naive CD4+ and Memory CD8+ T cells (Fig 2c right, Fig S4). We deduced that the traditional clustering algorithm based on Euclidean distance is difficult to fit irregularly distributed cells accurately.

![Figure 2](image)

**Figure 2** Evaluation of traditional clustering effectiveness after de-noising

a. UMAP visualization of cells without noise removal (Phase I; 13710 cells left), after removal of inferred pseudo-cells (Phase II; 11319 cells left), after removal of inferred impurity cells (Phase III; 10374 cells left), color-coded labels for pre-enriched cell types; b. The line plot of 4 class internal and 5 class external metrics for 5 clustering procedures on the 3-phase data; c. Part of the cluster distribution changed following noise cells stepwise removal, and clusters are labeled by the color of the largest proportion of cells and the color panel for cell types is the same as a.

**The consistency between different specific-gene analysis procedures**

We employed 5 specific-gene analysis procedures to reanalyze the specific genes for the 10 types of immune cells (Methods). The intersection of obtained specific-gene lists showed that t.test-, limma37-, and wilcox-based procedures had high consistency (Fig 3a part 2-3), while FC.rank (Fold Change) and LRT-based (Likelihood Ratio Test) procedures had low consistency with all other procedures (Fig 3a part 4-6). And the gene lists from the 3 high consistency methods showed more gene intersection with the prior collected specific-gene lists (Fig 3a part 1; Fig
S5; Table S8), which we then nominated as primary methods. Moreover, with the noise cell removal, the overall consistency of the 5 methods was improved (Fig. 3b), which was mainly demonstrated by significantly reduced non-intersection part (Fig. 3b left 1) and significantly increased co-intersection part (Fig 3b right 1).

**Figure 3** Consistency of specific gene lists obtained by different methods

**a.** The intersection of the specific gene lists obtained by the 5 methods and prior knowledge; **b.** The line plot of the intersection number of 5 methods along with the noise cells removal, and in each intersection case, the difference between the interaction on phase I data and phase III data was compared using the t-test, the change amount and the p-value were marked.

**Deep-LDA for single-cell classification and specific gene analysis**

By analogizing cell types as topics, and gene proportions as word frequencies, we applied an LDA-based deep generative model on scRNA-seq data analysis (Fig 4a; Methods). The model couples the optimization process of classification with class-specific gene analysis, and directly takes the count matrix as input, which greatly simplifies the analysis process and saves time (Fig 4b, Table S9). We applied the model on phase I data, and finally obtained 24 classes merged from the initial set 50 classes according to the class-specific genes intersection (Fig S6a; Methods). The annotated class distributions were in high concordance with the real cell distribution (Fig 4c). And the inferred specific genes showed high specificity for different classes (Fig 4d; Fig S6b). Then, we re-analyzed class-specific genes by the 5 procedures tested before, and intersected the outcomes with Deep-LDA specific gene lists. The intersection result indicated the specific genes of Deep-LDA had a high degree of consistency with that of the 3 primary methods nominated before (Fig 4d), which suggested Deep-LDA could also be a primary choice for specific gene analysis.
Figure 4 Architecture and Analysis Results of Deep-LDA

a. Schematic diagram of the model generation process and the corresponding variational model; b. Comparison of analytical procedure and time between Deep-LDA and traditional analysis methods; c. Clustering result of phase I data. The black numbers marked the position of the cluster centers (cluster index see Fig S6a). The clusters were colored by the cell type of the largest proportion, and the cell type color panel is the same as Fig 1b; d. Heatmap of the β matrix, the class ranking is the same as Fig S6a. The left color bar denotes the color of the cluster which is the same as e, and the values in parenthesis represent the purity; e. The intersections between the specific gene lists of Deep-LDA and that of the other 5 methods, and the quantitative ceiling indicated the maximum intersection number.

Deep-LDA had outstanding classification performance and lower noise sensitivity

Deep-LDA model was applied on the 3-phase data, whose clustering results had a high consistency with the real distribution at all phases (Fig 5a; Fig S7a). From the annotated memory CD8+ T cell distribution, we found that the distribution shape drawn from this model was more similar with the real distribution shape, and did not form a blocky distribution like other clustering procedures (Fig 5a right, 5b), which suggested Deep-LDA has a higher nonlinear fitting ability. Then, we applied the 13-clustering metrics to compare the Deep-LDA model with the other 5 clustering methods. And the external evaluation score lines of the Deep-LDA model were located above or near the upper lines of other methods’ borders, and exceeded more especially on the phase I part, which indicated Deep-LDA had an outstanding performance and should be one of the first echelon methods (Fig 5c; Fig S7b). In addition, the linear fitted slope of the evaluation score lines revealed that Deep-LDA had lower noise sensitivity than other methods (Fig 5c; Fig S7b). The poor internal clustering scores of Deep-LDA indicated that the outcome of the model was not optimized according to the uniform dimensionality reduction space which was the space for internal clustering metrics calculation, but was optimized according to the inferred feature space of different classes.
**Figure 5** Clustering results of Deep-LDA and the class external evaluation metrics

a. Deep-LDA clustering results on the three-phase data, and color annotated with the cell type of maximum occupancy cells; b. Memory CD8+ T cell distribution area and the same area of the other 3 traditional clustering procedures; c. The superposition of the bar and the line plot of the Deep-LDA clustering evaluation score. The light blue quadrilateral box is the outline of the line plots of the other 5 traditional methods, and the upper and lower sides are the fitting lines of the best method and the worst method, respectively. The $K_{DL}$ is the fitting slope based on the Deep-LDA, and the $K_{TW}$ is the average slope of the other 5 methods.

**Single-cell annotation performed by Deep-LDA transfer learning**

The encoder in Deep-LDA is for topic assignment, which serves as the cell type classifier. And the decoder is to generate the gene probability distribution, whose parameter was optimized to be class-specific gene frequency (Methods). Thus, we can preload the gene frequencies from independent reference datasets into the decoder, and then freeze the decoder in the parameter inference process, which can generate a single-cell classifier by transferred knowledge. In this project, we used the centroids of the 10 types of cells and an independent immune cell reference dataset as transferred knowledge (Fig 6a, b left), and then applied the optimized models to perform single-cell annotation (Methods). From the distribution on the UMAP plot, the annotated cell of both inferred results matched well with real related-cell distribution (Fig 6a, b right). After that, we compared the two results with the outcomes from cellID and singleR annotation tools with the 13-clustering metrics (Fig S8a, b; Methods), which verified the better performance of Deep-LDA transfer learning (Fig 6c; Fig S8c).
Figure 6 The transfer learning results of Deep-LDA and evaluation metric comparison

Deep-LDA used: a. centroids of the 10 types of cells and b. an independent reference dataset for transfer learning. The left panel shows the heatmap of pre-obtained $\beta$ matrix parameters, and the right panel shows the single-cell annotation results, marked with the number and purity for each cell type. c. Class evaluation score comparison of Deep-LDA’s annotation results with 2 other annotation methods.

Discussion

Clustering is the crucial and essential step in the current scRNA-seq data analysis workflow\textsuperscript{20}. In this paper, we mainly worked on clustering-related evaluation, optimization and redevelopment. Firstly, we applied scRNA sequencing on a pool of pre-labeled immune cells, by which a batch of data with main and fine labels was generated as ground truth for clustering evaluation. Then, we eliminated two types of noise cells inferred by deep generative models, which improved clustering performance and the consistency of several specific-gene analysis methods. After that, by applying the Deep-LDA model for scRNA-seq data analysis, which couples the classification process with class-specific genes analysis process, that is, the two processes are mutually dependent, we successfully simplified the analysis workflow, shortened the analysis time, and at last verified its superior clustering performance.

The prior knowledge of cell-specific genes is the basis for clustering annotation\textsuperscript{20}. However, the recommended reliable specific genes of the current databases were not ranked high and even not significant in our specific lists generated from the pooled sequencing data, and particularly worse in that of the fine-label clusters. More than that, the resolution that the scRNA-seq can achieve was determined by transcriptome features\textsuperscript{41}, and it should be much higher than the clusters classified by fine labels, which were defined by cell membrane proteins in our data. So, the current quality and quantity of the specific genes in prior knowledge were insufficient for accurate single-cell annotation\textsuperscript{42}. In addition, the quality of the obtained specific gene from most of the current scRNA-seq articles is unstable, because of the unstable clustering quality. Therefore, the more non-randomly captured or pre-labeled
single-cells are sequenced, due to the ground truth existence, the more cell-specific gene with higher reliability will be contributed to current databases.

The community-detection algorithm is the primary choice in the current clustering analysis, which was confirmed by the superior performance of louvain- and leiden-based methods in our project. However, an earlier applied method, the density peak clustering, which only grouped cells by the cell density on the two-dimension plot, can basically achieve the same superior performance. The reason is that the community-detected clusters also tend to form distinguishable distribution on UMAP/tSNE, which is the basis for density peak clustering. By specifically surveying the cell distribution on UMAP/tSNE, we found that for the more subdivided cell types, their shapes will be more irregular, and the boundaries between them will be more blurred. We deduced that this undistinguishable distribution will limit the capabilities of both community detection and density peak clustering. The major reason is that the feature space for classes with the different subdivided degrees is treated uniformly, which should have different granularity between inter-main label clusters and inter-fine label clusters. Therefore, we recommend that the initial clusters obtained should be merged into major clusters by similarity index. And then the high variance genes within the major cluster can be re-screened, that is, using finer-grained features of inner-cluster cells, to make a new round of clustering for fine clusters.

The positive contribution of inferred noise cells removal was confirmed in our study. In addition, unlike reshaping expression values for normalization, the excess quality control of noise cells, such as removing the cells in differentiated states, would also have a positive contribution. This deduction should be understood as that we should first map out the class relationships among the more discriminative cells in the rough-grained latent space, and then analyze the relationships of the cells of the less discriminative with these established broad classes. The impurity cell inference model in our project can learn the class affiliation probability. In the result part, the cells with higher impurity probability are always distributed in the junction between two clusters (Fig. S2b). Besides that, the deep generative model is an effective tool for estimating various types of noise cells, which can make a flexible assumption about how the noise cells are involved in the data generation, and then infers the noise probability of each cell by the observation data. In addition to the model inference which has to make an assumption, we found that noise cells had distinguishable distribution patterns on UMAP/tSNE (Fig 2a), so manually selecting abnormally distributed cells should also be considered a good way.

In the Deep-LDA model, the training parameter of specific gene frequency gradually showed different granularity inter- and inner-major clusters, which illustrated that the model is suitable for making classification in the data space of main and fine clusters. And the initial set parameter of the class number describes the resolution for classification. The generative architecture of Deep-LDA in our project was the classical LDA architecture of topic modeling and was not re-designed according to the characteristic of scRNA-seq data, such as incorporating the parameter for controlling the 0-inflation ratio. The neural network structure of the variational model was implemented by fully connected layers. In practice, we had tested CNN, RNN and slide layers in model construction, but they did not show better results. Therefore, by incorporating a more suitable network for learning the transcriptome data characteristics (functional groups and network structure, etc.) and incorporating more regulatory parameters in the generative process, the model will have greater potential in the future.

Methods

Participant. A healthy volunteer was enrolled in the study and signed the informed consent of sample collection and data analysis. The study was approved by the Ethics Committee of the Institute of Biomedical Sciences, Fudan University (No. 2018IBSJS014). The 150mL of peripheral blood was extracted by a professional doctor from the subject in the hospital laboratory, and the sample was anticoagulated by EDTA(Ethylene Diamine Tetraacetic Acid) and stored at 2~8°C.
**Immune cell enrichment.** Peripheral blood mononuclear cells (PBMCs) were isolated from the phase above the red blood cells by density gradient centrifugation using Lymphoprep™ (STEMCELL Technologies) as per the manufacturer’s instructions. The cells were counted via trypan blue stain and haemocytometer. Then, PBMCs were suspended to $5 \times 10^7$ cells/mL. The ten types of immune cells were enriched and purified by EasySep™ Human Series kits based on MicroBeads positive and negative screening (Table S1). The purification process was conducted strictly according to the manufacturer’s instructions. The purified and enriched cells were separated for subsequent single-cell sequencing and bulk sequencing (Figure 1a).

**Pooled sample scRNA-seq.** Each type of enriched and purified cell was mixed with a human universal antibody conjugated with Sample Tag (a specific oligonucleotide sequence) for sample identification, which has common 5' and 3' ends and the specific 'Sample Tag sequence', whose combination is as follows:

```
GTTGTCAAGATGCTACCGTTCAGAG | Sample Tag sequence | AA....25......AA
```

The relationship between cell type and 'Sample Tag sequence' is shown in Table S4. After the mixture, 10 types of labeled cells constituted a pooled sample for single-cell sequencing43. Single cells were captured by microwell plates and were lysed, and mRNA was hybridized with primer sequences on Cell Capture Beads. The BD Rhapsody™ Targeted mRNA and Abseq Amplification Kit were used for library preparation. The libraries were sequenced on Illumina’s sequencer. Information about Tag and WTA files was found in Table S5.

**Bulk sequencing.** 10 types of enriched and purified cells were respectively sampled (each sample has more than 100 cells) for bulk sequencing and repeated 3 times. A total of 30 samples were sequenced. The samples were lysed, reverse transcribed into cDNA, and amplified for library preparation. SMART-Seq® HT Kit was used for cDNA amplification. The quality of the cDNA library was evaluated, and 6 samples were excluded. Finally, 24 samples (Table S6) were sequenced on Illumina’s sequencer.

**Specific genes collection.** We collected the specific genes from the Cellmarker database44, PanglaoDB database45, and SC2Disease database46. The selected entries for each database are shown in Table S7. The relevant specific genes of immune cell47, T cell48, CD4+ T cell, CD8+ T cell49, and Naive T cell50 were obtained from references. The specific genes of Naive T cell were obtained from the intersection of Naive CD4+ T cell and Naive CD8+ T cell. The specific genes of B cell were obtained from the intersection of Naive B cell and Memory B cell. The specific genes appearing in two or more cell types were removed. Finally, a total of 460 specific genes for 16 cell types were collected (Table S8).

**Data processing.** Fastq files of the two types of sequencing were quality-checked with FASTQC, and adapters were removed by TrimGalore. For single-cell sequencing data, Cell barcode and UMI sequence were estimated and extracted by UMItools in the R1 files, and were then inserted into the R2 files. The R2 files were processed by STAR for single-end alignment. As for bulk-seq sequencing data, the alignment was paired-end. The reference genome was GRCh38(www.ncbi.nlm.nih.gov/assembly/GCF_000001405.39). The annotation was Genecode v.29(www.gencodegenes.org/human/release_29.html). Read count was adjusted by the correspondent UMIs. The true relationship between cell barcode and Sample Tag Sequence was estimated via Seurat’s Demultiplexing HTOs function (Demultiplexing with hashtag oligos) in Tag files43. Then, the count matrix was log-transformed. 2000 genes with the maximum variance were selected and scaled. The top 30 principal components (PCs) were obtained by PCA dimensional reduction. And the cell distribution was
visualized by UMAP and tSNE based on those PCs. For the 3-phase data with a stepwise noise cells removal, the data processing was the same.

**The composition of bulk-seq data.** The 2000 genes from the data processing were used as gene signatures for deconvolution. And then, the centroids of 10 types of immune cells were calculated to be the Signature gene file. Then, we used CIBERSORT software to decompose the bulk-seq data by the signature gene file and to verify the consistency between scRNA-seq data and bulk-seq data.

**Clustering Analysis.** A total of five algorithms, densitypeak clustering, DDRTree (Discriminative dimensionality reduction via learning a tree) clustering, louvain clustering, leiden clustering, and consensus clustering, were used for clustering. Densitypeak clustering and DDRTree clustering were implemented by monocle (version 2.22.0). The tSNE coordinate was used for densitypeak clustering. The DDRTree dimension-reduced data was used for DDRTree clustering. The clustering was implemented by the clusterCell function of monocle with ‘method’ equal to ‘densityPeak’ or ‘DDRTree’ . Louvain clustering and leiden clustering were implemented by monocle3 (version 1.0.0) with PCs (dim=30), the cluster_cells function with ‘method’ equal to ‘louvian’ or ‘leiden’. Consensus clustering was implemented by the scena_cpu function of SCENA (version 1.0.3). The cluster number was set as 18, which led to the optimal results.

**Specific genes analysis.** The count matrix was normalized as recommended. The pairwise comparison strategy was used for specific genes analysis, where the core differential analysis algorithms were by t-test, limma, wilcox, logFC ranking, and LRT (likelihood ratio test). The analysis based on the t-test was implemented by the findMarkers function of scran (version 1.22.1). Limma used the one-VS-others pairwise strategy for all clusters. The analysis based on wilcox was implemented by the FindAllMarkers function of Seurat (version 4.0.5). The analysis based on LogFC ranking was implemented by the getClassicMarkers function of singleR (version 1.8.1). The analysis based on the LRT was implemented by the differentialGeneTest function of monocle (version 2.22.0). Then, for the differential gene lists of each cell type obtained from the different analyses, the 250 top-ranking genes were selected as specific genes.

**Cell Type Automatic Annotation.** SingleR39 (version 1.8.1) and CellID39 (version 1.4.0) were used for cell type automatic annotation. The reference data with known cell types was used to annotate experimental data, where HumanPrimaryCellAtlas and PanglaoDB_markers_27_Mar_2020 were, respectively, the reference data of SingleR and CellID.

**Intra-cluster Evaluation Index.** Silhouette Coefficient index was a measure of the cohesion and separation of clusters and tended to be higher, when the distance between samples of the same type was smaller and between samples of the different types was larger. The formula was the following:

$$SC = \frac{N}{\sum_{i=1}^{n} \frac{d_i - \bar{s}_i}{\max(s_i, d_i)}}$$

Where N denoted the number of all samples. $s_i$ denoted the average distance between sample i and all the other samples with the same cell type label. $d_i$ denoted the average distance between sample i and all the other samples with the nearest and different sample type labels. The score was evaluated via metrics.silhouette_score function of sklearn.

Calinski-Harabaz index is a measure of cohesion via covariance. The intra-cluster is closer and the inter-cluster
is more scattered, so the score is higher. The formula was the following:

\[ SS_q = \sum_{q=1}^{k} n_q (c_q - c_z)(c_q - c_z)^T \]
\[ SS_d = \sum_{q=1}^{k} \sum_{x \in V_q} (x - c_q)(x - c_q)^T \]
\[ CH = \frac{SS_q N - k}{SS_d k - 1} \]

Where \( N \) denoted the number of all samples, \( k \) denoted the number of clusters, \( n_q \) was the number of samples within cluster \( q \), \( c_q \) was the centroid of all samples, \( V_q \) was the set of cluster \( q \). \( SS_d \) was the sum of the covariance of each cluster centroid, \( SS_q \) was the sum of covariance of the sample within the same cluster. The value was calculated by metrics.calinski_harabaz_score function of sklearn.

Compactness index was the average distance between the data point and the centroid of the cluster. The formula was the following:

\[ CP = \frac{1}{k} \sum_{q=1}^{k} \frac{1}{n_q} \sum_{x \in V_q} \| x - c_q \|_2 \]

Where the meaning of parameters unexplained was identical with those described above.

Davies-Bouldin index was to calculate the averaged inner-cluster distance of 2 clusters, which was divided by the distance of the centroids of 2 clusters, and then found the maximum. The score tended to be lower when the inner-cluster distance was smaller, and the intra-cluster distance was larger. The formula was the following:

\[ DB = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \{ \frac{\sum_{x \in V_i} (x - c_i) \cdot (x - c_j)}{\| c_i - c_j \|_2} \} \]

Where the meaning of parameters unexplained was identical with those described above. The value was calculated by metrics.davies_bouldin_score of sklearn.

**Inter-cluster Evaluation Index**

Purity was calculated as the following formula:

\[ Pur = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \{ \sum_{p_i \cap r_j} \} \]

Where \( p_i \) was the sample set of clusters \( i \), \( r_j \) was the sample set of clusters \( j \), and to obtain the maximum number of intersections between \( p_i \) and \( r_j \).

Rand index was the ratio of true prediction and was calculated as the following formula:

\[
\begin{align*}
    TP &= \sum_{p_i \cap r_j} \delta(p_i, r_j) \\
    TN &= \sum_{p_i \cap r_j} \delta(p_i, r_j) \text{ For } (i,j) \in \text{all pairs in } V \\
    FP &= \sum_{p_i \cap r_j} \delta(p_i, r_j, i \neq j) \\
    FN &= \sum_{p_i \cap r_j} \delta(p_i, r_j, i = j) \\
    \end{align*}
\]

\[ RI = \frac{TP + TN}{TP + FP + FN} = \frac{TP + TN}{\binom{N}{2}} \]

Where \( V \) was the set of all samples, \( p_i \) was the predicted cell type of sample \( i \), \( r_i \) was the true cell type of sample \( i \), \( n \) was the sample number, \( \binom{N}{2} \) was the combination calculation.

Adjusted Rand index was the standardized Rand index, and the formula was the following:

\[
ARI = \frac{RI - E(RI)}{\max(RI) - E(RI)} = \frac{\sum_{i,j} C_{ij} - (\sum_i C_{ij}^2 / C_k^2) - (\sum_j C_{ij}^2 / C_k^2)}{\frac{1}{2} (\sum_i C_{ii}^2 + \sum_j C_{jj}^2) - (\sum_i C_{ij}^2 + \sum_j C_{ji}^2) / C_k^2}
\]
Where $\Omega_{i,j}$ was the intersection number of predicted cluster $i$ and true cluster $j$. $a_i$ was $\sum_j \Omega_{i,j}$ and $b_j$ was $\sum_i \Omega_{i,j}$.

The value was calculated by metrics.adjusted_rand_score of sklearn.

F-score was defined as:

\[
\begin{align*}
\text{Precision} &= \frac{TP}{TP + FP} \\
\text{Recall} &= \frac{TP}{TP + FN} \\
F_{\beta^2} &= \frac{(1 + \beta^2) \text{Precision} \times \text{Recall}}{\beta^2 \text{Precision} + \text{Recall}}
\end{align*}
\]

Where the meaning of parameters unexplained was identical with those described above. $\beta$ was the hyperparameter and was set to 1.

Fowlkes-Mallows index was the geometric mean of pair-wise precision and recall.

\[
FMI = \frac{TP}{\sqrt{(TP + FP)(TP + FN)}}
\]

Where the meaning of parameters unexplained was identical with those described above. The value was calculated by metrics.fowlkes_mallows_score function of sklearn.

AMI (Adjusted mutual information) index:

\[
\begin{align*}
H(U) &= \sum_{i=1}^{u} P(i) \log(P(i)) \\
P(i) &= \frac{U_i}{N} \\
MI(U, V) &= \sum_{i=1}^{u} \sum_{j=1}^{v} P(i, j) \log \left( \frac{P(i, j)}{P(i)P(j)} \right) \\
P(i, j) &= \frac{U_i \cap V_j}{N} \equiv \Omega_{i,j}/N \\
AMI &= \frac{MI - E(MI)}{\max(H(U), H(V)) - E(MI)} \\
E[MU(V)] &= \sum_{i=1}^{u} \sum_{j=1}^{v} \min(U_i, V_j) \frac{m}{N} \frac{N \times m}{U_i \times V_j} \frac{N!}{U_i!V_j!} \frac{m!}{(U_i - m)!V_j!(m!)} \frac{(N - U_i)!(N - V_j)!}{(N - U_i - V_j + m)!}
\end{align*}
\]

Where the meaning of parameters unexplained was identical with those described above. $U$ and $V$ represented the predicted cluster and true cluster, $P(i)$ was the proportion of cells labeled $i$, $P(i, j)$ was the proportion of the cells of predicted label $i$ and true label $j$. The value was calculated by metrics.adjusted_mutual_info_score of sklearn.

Homogeneity measured whether all clusters fell into the same clusters with the true label. The formula was the following:

\[
\begin{align*}
H(U|V) &= -\sum_{i=1}^{u} \sum_{j=1}^{v} P(i, j) \log(P(i, j)) \\
HO &= 1 - \frac{H(U|V)}{H(U)}
\end{align*}
\]

Where the meaning of parameters unexplained was identical with those described above. The index was calculated by metrics.homogeneity_score of sklearn.

Completeness measured the extent to which samples belonging to the true same cluster fell into the predicted same cluster. The formula was the following:

\[
\begin{align*}
CO &= 1 - \frac{H(V|U)}{H(V)}
\end{align*}
\]
The value was calculated by metrics.completeness_score of sklearn.

\[ V = \frac{2 \cdot HO \times CO}{HO + CO} \]

The value was calculated by metrics.v_measure_score function of sklearn.

**Pseudo cell generative model.** The model was constructed according to the structure of cellbender\textsuperscript{27}, and the observed data \( X_{ng} \) was generated as follows:

\[
\begin{align*}
&z_n \sim \text{Normal}(0,1) \\
&c_{ng} \sim \text{Normal}(c_n) \\
&d_{n}^{\text{true}} \sim \text{lognormal}(d_n^{\text{true}}, d_n^{\text{true}}) \\
&d_n^{\text{fake}} \sim \text{lognormal}(d_n^{\text{fake}}, d_n^{\text{fake}}) \\
&q_n \sim \text{Gamma}(\phi_n, \phi_n) \\
&y_n \sim \text{Bernoulli}(p) \end{align*}
\]

\[
\begin{align*}
&\mu_{ng} = \gamma_n \times d_n^{\text{true}} \times c_{ng} + d_n^{\text{empty}} \times A_g \\
&\theta_g = 1/\phi_g \\
&X_{ng} \sim \text{ZINB}(\mu_{ng}, g_{ng}, \theta_g)
\end{align*}
\]

Where \( z_n \) was the latent variable, which was hypothetically generated according to the normal distribution. During the inference process, an encoder (encoder_z) was used as a variational model with the input of gene counts for generating the mean and variance of \( z_n \). \( c_{ng} \) was the generated true gene expression values derived from latent space through an decoder (\( NN_j \)). \( g_{ng} \) was the probability that gene was not observed. \( d_n^{\text{true}} \) was the scale factor of the true cell, and \( d_n^{\text{cell}} \) was generated by encoder_d in the inference process. \( d_n^{\text{empty}} \) was the scale factor of the pseudo cell. \( y_n \) was the label of whether the cell was a true cell, where the probability was generated by endoder_p in the inference process. \( \mu_{ng} \) was the generated true gene expression values after combining the hypothesis of the pseudo cell, where \( A_g \) was the ambient RNA. \( \phi_g \) was inverse dispersion parameter of the model. The final observation value \( X_{ng} \) was generated by ZINB (Zero-inflated Negative Binomial) model. The parameter priors of the generation process are given in Table S10.

**Impurity cell generative model.** There exited impurity cells in each cell cluster, because the cells’ purification degree was less than 100% (Table S2). The class affinity probability for each cell was inferred by the semi-supervised generative model\textsuperscript{51} proposed by scANVI\textsuperscript{52}. The observed data \( X_{ng} \) was generated as following:

\[
\begin{align*}
&z_n \sim \text{Normal}(0,1) \\
&\begin{cases} 
\text{if label is True: } y_n = \text{label} \\
\text{else: } y_n = \text{OneHotCategorical}(p_i)
\end{cases} \\
&w_n^e, w_n^c \sim \text{Normal}(w_n, w_n) \\
&w_n \sim \text{Normal}(w_n, w_n) \\
&d_n \sim \text{LogNormal}(d_n, d_n) \\
&c_{ng} \sim \text{Normal}(c_n) \\
&q_n \sim \text{Gamma}(\phi_n, \phi_n) \\
&\mu_{ng} = d_n \times c_{ng} \\
&\theta_g = 1/\phi_g \\
&X_{ng} \sim \text{ZINB}(\mu_{ng}, g_{ng}, \theta_g)
\end{align*}
\]

Where the meaning of \( z_n, c_{ng}, g_{ng}, \mu_{ng} \) and \( \phi_g \) was the same as in the Pseudo Cells Inference Model. \( w_n \) was generated by a decoder with latent variable \( z_n \) and cell label \( y_n \) as input. Because Impurity Cells Inference Model
was a semi-supervised model, we set 50% of cell labels were unknown. In the unknown case, the cell label $y_n$ was sampled from \texttt{OneHotCategorical} distribution and the parameter $p_1$ was the probability of the cell label generated by encoder $p$ in inference process. The parameter priors of the generation process are given in Table S10.

### Generative model inference.

The posterior parameter distribution inference was combined with the prior distribution and observed data $X$. And because the $p(\lambda_1 \cdots \lambda_k | X_n)$ can hardly be deduced, we used the variation inference to infer the approximately posterior distribution. Variation distribution can be divided into independent distributions of different parameters. The formula was as follows:

$$q(\lambda_1 \cdots \lambda_k | X_n) = \prod_{i=1}^{k} q(\lambda_i | X_n)$$

The likelihood function was deduced as:

$$\log p(X | \theta_1 \cdots \theta_k) \geq E_{\theta_1 \cdots \theta_k} p(X | \theta_1 \cdots \theta_k) - \sum_{i=1}^{k} D_{KL}(q(\lambda_i | X_n) \parallel p(\lambda))$$

The evidence lower bound (ELBO) was set as the optimization objective. The optimization was realized by \texttt{pyro}. The loss models and the optimizer for the three models described above were ELBO and Adam.

### Gene selection for the deep generative model.

Genes that are expressed or not expressed in the vast majority of cells are more useless in distinguishing cell types, so we select genes expressed in a certain percentage of cells for the optimization of the deep generative model. This project used genes expressed in cells that account for 40~90%, a total of 2441 genes.

### Deep-LDA model.

This model is an LDA-based (Latent Dirichlet Allocation) deep generative model. And the generation structure was referenced from proLDA\textsuperscript{30}. The gene frequency was analogized as word frequency, and the cell type distribution was analogized as the topic distribution. The observed data $X_{ng}$ was as follows:

$$\theta \sim \text{LogNormal}(\theta_{\mu}, \theta_{\sigma})$$

$$p_{ng} = NN(\theta)$$

$$X_{ng} \sim \text{Multinormal}(N, p_{ng})$$

Where $\theta$ denoted the cell type distribution, which should be generated by the Dirichlet distribution and here generated by a lognormal distribution. In the inference process, $\theta$ was generated by an encoder (variation model) with the input of the gene counts. $p_{ng}$ denoted the gene probability distribution, which was generated by a decoder ($NN_\lambda$) and could be represented as $p_{ng} = \text{softMax}(\beta \times \theta)$. So, the weight $\beta$ can be used as the training result of gene frequency for different topics or cell types. The observation value $X_{ng}$ was generated by the \texttt{Multinormal} distribution and $N$ was all genes count of a cell. The parameter priors of the generation process are given in Table S10.

### Specific gene of Deep-LDA.

The optimized $\beta$ matrix ($N_{\text{class}} \times N_{\text{gene}}$) in the generative model was used as the basis for specific gene screening. That is, for the optimized gene frequency list of each class, a certain number (default is 250) of high-frequency genes were selected as specific genes for the class.

### Class merging of Deep-LDA.

Based on the specific gene intersections of different classes obtained by Deep-LDA, a threshold was set and classes with the number of intersected genes above this threshold were merged. In this process, we suggest to select more specific genes for each class (for phase I ~ phase III data, the number of specific genes used for merging classes are set as: 1700, 1400, 600), and set a certain threshold (for phase I ~ phase III data,
the thresholds for merging are: 1500, 1000, 300) to produce the merged cluster number as expected (for phase I ~ phase III data, the number of clusters are: 24, 18, 21).

Transfer learning of Deep-LDA. We first intersected the gene list of the signature matrix (transferred knowledge) with that of the data to be analyzed so that both matrixes contained the same gene signatures. Here, we used two types of signature files. The first one being the centroid of each cell type through our pooled sample sequencing data, and the 2441 genes for the deep generative model served as gene features. The second one is obtained independently from the human genome array (GSE22886)38, containing a total of 22 cell types, and the gene features are selected according to the 530 specific genes supplied by CIBERSORT31. Then, the signature matrix (count or raw array intensity data type) was scaled and loaded into the β parameter of the decoder. Deep-LDA was then performed on inference with the data to be analyzed while the decoder parameters were fixed. The encoder obtained after optimization was used as a classifier for annotating new data using transferring knowledge.

Reference

1. Neu KE, Tang Q, Wilson PC, Khan AA. Single-Cell Genomics: Approaches and Utility in Immunology. Trends Immunol 38, 140-149 (2017).
2. Ofengeim D, Giagtzoglou N, Huh D, Zou C, Yuan J. Single-Cell RNA Sequencing: Unraveling the Brain One Cell at a Time. Trends Mol Med 23, 563-576 (2017).
3. Gohil SH, Iorgulescu JB, Braun DA, Keskin DB, Livak KJ. Applying high-dimensional single-cell technologies to the analysis of cancer immunotherapy. Nat Rev Clin Oncol 18, 244-256 (2021).
4. Ren X, Zhang L, Zhang Y, Li Z, Siemers N, Zhang Z. Insights Gained from Single-Cell Analysis of Immune Cells in the Tumor Microenvironment. Annu Rev Immunol 39, 583-609 (2021).
5. Sun D, et al. Identifying phenotype-associated subpopulations by integrating bulk and single-cell sequencing data. Nat Biotechnol 40, 527-538 (2022).
6. Gonzalez H, et al. Cellular architecture of human brain metastases. Cell 185, 729-745.e720 (2022).
7. Breda J, Zavolan M, van Nimwegen E. Bayesian inference of gene expression states from single-cell RNA-seq data. Nat Biotechnol 39, 1008-1016 (2015).
8. Lun AT, McCarthy DJ, Marioni JC. A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. F1000Res 5, 2122 (2016).
9. Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol 36, 421-427 (2018).
10. Yu T. A new dynamic correlation algorithm reveals novel functional aspects in single cell and bulk RNA-seq data. PLoS Comput Biol 14, e1006391 (2018).
11. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods 15, 1053-1058 (2018).
12. Risso D, Perraudeau F, Gribkova S, Dudoit S, Vert JP. A general and flexible method for signal extraction from single-cell RNA-seq data. Nat Commun 9, 284 (2018).
13. Hafemeister C, Satija R. Normalization and variance stabilization of single-cell RNA-seq data using regularized negative binomial regression. Genome Biol 20, 296 (2019).
14. Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data.
17. Levine JH, et al. Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells that Correlate with Prognosis. *Cell* **162**, 184-197 (2015).

18. Waltman L, van Eck NJ. A smart local moving algorithm for large-scale modularity-based community detection. *The European Physical Journal B* **86**, 471 (2013).

19. Young MD, Behjati S. SoupX removes ambient RNA contamination from droplet-based single-cell RNA sequencing data. *Gigascience* **9**, (2020).

20. Andrews TS, Kiselev VY, McCarthy D, Hemberg M. Tutorial: guidelines for the computational analysis of single-cell RNA sequencing data. *Nat Protoc* **16**, 1-9 (2021).

21. Zappia L, Theis FJ. Over 1000 tools reveal trends in the single-cell RNA-seq analysis landscape. *Genome Biol* **22**, 301 (2021).

22. Lähnemann D, et al. Eleven grand challenges in single-cell data science. *Genome Biol* **21**, 31 (2020).

23. Baruzzo G, Patuzzi I, Di Camillo B. SPARSim single cell: a count data simulator for scRNA-seq data. *Bioinformatics* **36**, 1468-1475 (2020).

24. Zhang X, Xu C, Yosef N. Simulating multiple faceted variability in single cell RNA sequencing. *Nat Commun* **10**, 2611 (2019).

25. Eraslan G, Simon LM, Mircea M, Mueller NS, Theis FJ. Single-cell RNA-seq denoising using a deep count autoencoder. *Nat Commun* **10**, 390 (2019).

26. Wang D, Gu J. VASC: Dimension Reduction and Visualization of Single-cell RNA-seq Data by Deep Variational Autoencoder. *Genomics Proteomics Bioinformatics* **16**, 320-331 (2018).

27. Fleming SJ, Marioni JC, Babadi M. CellBender remove-background: a deep generative model for unsupervised removal of background noise from scRNA-seq datasets. *bioRxiv*, 791699 (2019).

28. Peng J, Xue H, Wei Z, Tuncali I, Hao J, Shang X. Integrating multi-network topology for gene function prediction using deep neural networks. *Brief Bioinform* **22**, 2096-2105 (2021).

29. Alghamdi N, et al. A graph neural network model to estimate cell-wise metabolic flux using single-cell RNA-seq data. *Genome Res* **31**, 1867-1884 (2021).

30. Akash Srivastava CS. Autoencoding Variational Inference For Topic Models. *arXiv*, 1703.01488 (2017).

31. Newman AM, et al. Robust enumeration of cell subsets from tissue expression profiles. *Nat Methods* **12**, 453-457 (2015).

32. Becht E, et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol*, (2018).

33. Launers VDM, Hinton G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9**, 2579-2605 (2008).

34. Ester M, Kriegel HP, Sander J, Xiaowei X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Conference: 2. international conference on knowledge discovery and data mining, Portland, OR (United States), 2-4 Aug 1996; Other Information: PBD: 1996; Related Information: Is Part Of Proceedings of the second international conference on knowledge discovery & data mining; Simoudis, E.; Han, J.; Fayyad, U. [eds.]; PB: 405 p.). AAAI Press, Menlo Park, CA (United States) (1996).

35. Qiu X, et al. Reversed graph embedding resolves complex single-cell trajectories. *Nat Methods* **14**, 979-982 (2017).

36. Cui Y, Zhang S, Liang Y, Wang X, Ferraro TN, Chen Y. Consensus clustering of single-cell RNA-seq data by enhancing network affinity. *Brief Bioinform* **22**, (2021).

37. Phipson B, Lee S, Majewski IJ, Alexander WS, Smyth GK. ROBUST HYPERPARAMETER ESTIMATION PROTECTS AGAINST HYPERVARIABLE GENES AND IMPROVES POWER TO
DETECT DIFFERENTIAL EXPRESSION. Ann Appl Stat 10, 946-963 (2016).

38. Abbas AR, et al. Immune response in silico (IRIS): immune-specific genes identified from a compendium of microarray expression data. Genes Immun 6, 319-331 (2005).

39. Cortal A, Martignetti L, Six E, Rausell A. Gene signature extraction and cell identity recognition at the single-cell level with Cell-ID. Nat Biotechnol 39, 1095-1102 (2021).

40. Aran D, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. Nat Immunol 20, 163-172 (2019).

41. Chen G, Ning B, Shi T. Single-Cell RNA-Seq Technologies and Related Computational Data Analysis. Front Genet 10, 317 (2019).

42. Travaglini KJ, et al. A molecular cell atlas of the human lung from single-cell RNA sequencing. Nature 587, 619-625 (2020).

43. Stoeckius M, et al. Cell Hashing with barcoded antibodies enables multiplexing and doublet detection for single cell genomics. Genome Biol 19, 224 (2018).

44. Zhang X, et al. CellMarker: a manually curated resource of cell markers in human and mouse. Nucleic Acids Res 47, D721-d728 (2019).

45. Franzén O, Gan LM, Björkegren JLM. PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. Database (Oxford) 2019, (2019).

46. Zhao T, et al. SC2disease: a manually curated database of single-cell transcriptome for human diseases. Nucleic Acids Res 49, D1413-d1419 (2021).

47. Rheinländer A, Schraven B, Bommhardt U. CD45 in human physiology and clinical medicine. Immunol Lett 196, 22-32 (2018).

48. Dong D, et al. Structural basis of assembly of the human T cell receptor-CD3 complex. Nature 573, 546-552 (2019).

49. Kumar BV, Connors TJ, Farber DL. Human T Cell Development, Localization, and Function throughout Life. Immunity 48, 202-213 (2018).

50. Besedovsky L, Linz B, Born J, Lange T. Mineralocorticoid receptor signaling reduces numbers of circulating human naïve T cells and increases their CD62L, CCR7, and CXCR4 expression. Eur J Immunol 44, 1759-1769 (2014).

51. Diederik P. Kingma DJR, Shakir Mohamed, Max Welling. Semi-Supervised Learning with Deep Generative Models. arXiv, 1406.5298 (2014).

52. Xu C, Lopez R, Mehmane E, Regier J, Jordan MI, Yosef N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. Mol Syst Biol 17, e9620 (2021).

Data availability:
The project contains a total of 1 pooled sample scRNA-seq data and 24 bulk sequencing data, which have been uploaded to SRA. The samples and uploaded data information are shown in https://dataview.ncbi.nlm.nih.gov/object/PRJNA843175?reviewer=s1iefbplretvm0005bsj2toce5o. The data will be available with the publication of this article.

Code availability
The three deep generative models covered in this article, as well as necessary clustering evaluation tools and plotting tools are organized into the ScDeepTools (https://github.com/liuyunho/ScDeepTools) software. In addition, the software includes an interactive cell distribution plotting tool (Fig S9) for manually cell picking and deletion.
Author contributions:
Conceptualization, Yunhe Liu, Gang Liu and Lei Liu; experiment, Yunhe Liu, Qiqing Fu, Chenyu Dong, Xiaoqiong Xia; methodology, Yunhe Liu, Qiqing Fu; software, Yunhe Liu; validation, Yunhe Liu, Qiqing Fu; manuscript preparation, Yunhe Liu, Qiqing Fu, Chenyu Dong; visualization, Yunhe Liu, Qiqing Fu, Chenyu Dong; funding acquisition, Lei Liu. All authors have read and agreed to the published version of the manuscript.

Funder:
L.L. was supported by the grant No. 2021YFC2500403 from the National Key Research and Development Program of China.

Acknowledgements:
For technical advice and financial assisting, we thank the project of Shanghai’s Double First-Class University Construction and Development of High-Level Local Universities: Intelligent Medicine Emerging Interdisciplinary Cultivation Project, and also Medical Research Data Center of Fudan University.