A Meta-learning based Graph-Hierarchical Clustering Method for Single Cell RNA-Seq Data

Zixiang Pan
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China
panzx6@mail2.sysu.edu.cn

Yuefan Lin
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China
linyf63@mail2.sysu.edu.cn

Haokun Zhang
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China
zhanghk25@mail2.sysu.edu.cn

Yuansong Zeng
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China
zengys@mail2.sysu.edu.cn

Weijiang Yu*
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China
weijiangyu8@gmail.com

Yuedong Yang*
School of Computer Science and Engineering
Sun Yat-Sen University
Guangzhou, China
yangyd25@mail.sysu.edu.cn

Single cell sequencing techniques enable researchers view complex bio-tissues from a more precise perspective to identify cell types. However, more and more recent works have been done to find more detailed subtypes within already known cell types. Here, we present MeHi-SCC, a method which utilized meta-learning protocol and brought in multi scRNA-seq datasets' information in order to assist graph-based hierarchical sub-clustering process. In result, MeHi-SCC outperformed current-prevailing scRNA clustering methods and successfully identified cell subtypes in two large scale cell atlas. Our codes and datasets are available online at https://github.com/biomed-AI/MeHi-SCC

Keywords—Single Cell RNA-seq, Hierarchical Clustering, Meta-learning, Cell Sub-type detection

I. Introduction

Single cell level sequencing was one of the most promising bio-technologies of this decade. It enables deep insight into cell-level biological process including cell communication and other micro-environment events[1, 2]. Nowadays, thanks to the progress of bio-engineering development and the next generation sequencing (NGS), single cell sequencing has already enabled us to detect signals via transcriptomes, DNA methylation and surface protein via micro-array chips[3, 4]. Among them, single cell RNA sequencing can be the most prevailing technology as it manifests more genetics information than other omics and have higher resolution compared to single cell DNA-seq with high performance-price ratio[5]. After fulfillment of Human Genome Project (HGP), larger scale cell atlas schedules were conducted like T cell atlas and B cell atlas, and quantity of cells were extended from 10^3 to 10^6[6-8]. However, technique limitations like sequence depth and systematic bias usually led to noise thus lower RNA counts sequencing accuracy like dropout[9]. At the same time, High dimension brought in challenges in dealing with batch effect and other differences between sequencing techniques, considering sometimes multi-source datasets are brought in for reference[10].

One primary task for single cell analysis is cell clustering. For now, single cell clustering helps detect new subtypes within known cell types, which helps reveal bio-mechanism in oncology micro-environment heterogeneity and tissue cell differentiation in embryo genesis[11, 12]. To tackle problems mentioned above, dimension reduction methods like LSI were developed[13], and low-dimension visualization methods like T-SNE, and UMAP are helpful for providing researchers a direct version into these high-redundant and high-dimensional datasets[14, 15].

Representation learning, aroused by dimension reduction, is now an important realm in analyzing single cell data in artificial intelligence, especially in coping with problems like imputation, cell type annotation and clustering[16-19]. Eraslan et al. first applied auto-encoder to extract deep low-dimensional space for single cells with Zero-inflated Negative Binomial distribution[9], while Tian et al. first used learnt latent space to generate clustering labels for further self-supervision with auto-encoder[20]. Xie et al. utilized dual self-supervised techniques to assign high dimension samples with several clustering centers[21], to better preserve local data feature structure, Guo et al. proposed IDEC, which preserved auto-encoder structure in dual self-supervision training phase[22]. To better reveal inter-cellular relationship, Zeng et al. integrated cell-cell relationship with graph neuron network as extra information to optimize dual self-supervised clustering[23]. In addition, Yi et al. applied graph attention network enhancement for the same goal and Hui et al. brought in contrastive learning to promote robustness[24, 25]. As for traditional machine learning based single cell clustering method, Seurat uses log-normalization to impute raw RNA counts and generate clustering labels based on nearest neighbors with high variant gene markers[26]. Wang et al. applied multi-kernel learning for better cell representation learning[27], while Lin et al. promoted principal component analysis and imputation, and is scalable to cluster thousands of cells[28].

However, traditional self-supervised clustering methods have limitations considering data sparsity and dataset specificity. One way to cope with the problem is to bring in more bio-medical information or multi-datasets as reference, and meta-learning is a brand new field to do so. Meta-Learning, namely "learning to learn", is a brand new concept based on large data and multi-task training, which was considered as a substitution with traditional fine-tuning, and now it has gained preference in many tasks in computer vision and nature language processing realms[29-31].

In this paper, we proposed a Meta-learning based Hierarchical Single Cell Clustering tool called MeHi-SCC.

* Corresponding Author

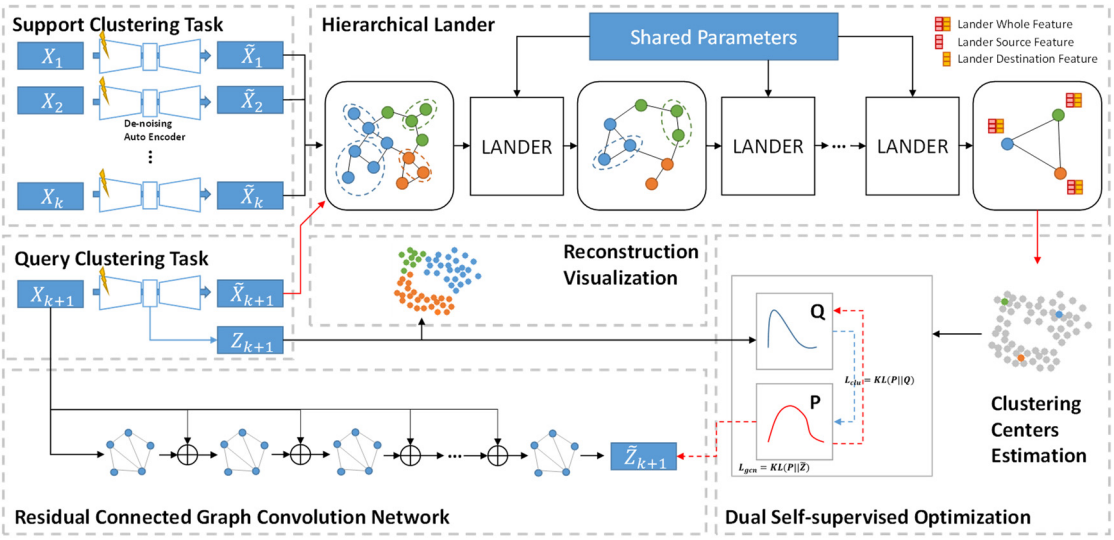

Fig.1. Basic Framework of MeHi-SCC.

The main motivation of MeHi-SCC is to explore a potential way to cluster a certain scRNA dataset into more detailed sub-types with the aid of hierarchical clustering. Inspired by Xing et al.[32], we used meta-training and brought in multi clustering tasks with multi-source scRNA datasets and their former manually annotated cell types as support tasks to promote clustering accuracy.

MeHi-SCC have three main innovating contributions as follows:

(1) MeHi-SCC features meta-learning. We trained an independent hierarchical clustering separator section LANDER with support clustering tasks[32]. Unseen scRNA datasets from different sequencing techniques, tissues or even different species were fed to LANDER one by one for training with annotated ground truth. Noticeably, support datasets have no overlap with query dataset.

(2) MeHi-SCC features a whole-graph-tuning based hierarchical clustering section. LANDER, the separator, only learns how inter-cellular relationship helps cluster step by step toward ground truth, ignoring specific expression values. Different from graph convolution neural-network(GNN) with fixed adjacent matrix, LANDER updates both edge-connections and related node features.

(3) MeHi-SCC enables sub-cell-type detection. Hierarchical LANDER helps divide cell graphs into sub-cell graphs and aggregate them into more detailed clusters for all cells until they cannot be divided into sub-graphs any more, and our cluster number is usually more than ground truth given by manual annotations from morphology or biomarkers.

In this work, we proved that MeHi-SCC performed better than current-prevailing benchmark pipelines. Besides, MeHi-SCC sees better latent representation visualization performance compared with other pipelines and we successfully identified cell sub-types in two large scale single cell atlas, from both human and mouse tissue.

II. Materials and Methods

MeHi-SCC mainly separates into three parts(Fig.1), i.e. de-noising auto-encoder part, meta-learning part, and dual sself-supervision part. All the parts are based on Pytorch framework[33]. De-noising auto-encoder part, as shown on the left-top of the figure, used for data pre-processing for both support and query datasets, is actually a set of auto-encoders with Gaussian noise added. Meta-training part, as shown on the right-top of the figure, is hierarchical LANDER, a meta-learning based hierarchical clustering separator machine.

Datasets should first be reconstructed by the de-noising auto-encoder. And then reconstructed expression profiles will be fed to well pre-trained LANDER as input to generate initial clustering results(red arrows).

Finally the dual self-supervision part, on the bottom of the figure, will calculate clustering centers from clusters generated by LANDER, and then latent spaces generated by the de-noising auto-encoder and the residual connected graph convolution network (Left Down) will approach to the distribution of clustering centers through Kullback-Leibler Divergence[34].

A. Data Preparation and De-Noising Auto-Encoder

We here prepared 15 open-sourced datasets for experiments, with 13 normal scale datasets and 2 large-scale datasets(PBMC 68K and Shekhar Mouse Retina). All the datasets have their manual annotated cell types, we downloaded their expression profiles and treated the cell type annotations as ground truth(Supplementary Materials Table S1).

Before scRNA-seq profiles were fed to the model, we first applied raw counts (including the query dataset and support training datasets) with TPM normalization via Seurat[26]. And then we selected 2000 High Variant Genes (HVGs) by default. Then the 2000-dimension-profile will be fed to the de-noising auto-encoder by 400 epochs and then to the LANDER.

Here, the origin scRNA profile $X \in \mathbb{R}^{N\times m}$ has N cells and m selected genes, and Gaussian noise Δ has the same dimension with X. The auto-encoder h consists of encoder h_e with dimensions $[m, 512, 256, 64, z]$ where z denotes latent dimension. $Z = h_e(X + \Delta)$ and $Z \in \mathbb{R}^{N\times z}$ is the latent space representation which is used for visualization, and then the latent space will be used as input of decoder h_d with dimensions $[z, 64, 256, 512, m]$ to calculate reconstruction expression profile $\tilde{X} = h_d(Z)$, $\tilde{X} \in \mathbb{R}^{N\times m}$. The optimization

goal for de-noising auto-encoder is mean square error(MSE), i.e. $MSE(X, \tilde{X})$.

B. Training the Meta Learning LANDER

LANDER is a Graph Convolution Network (GNN) based framework with two part —— the separator ϕ and aggregator ψ. The separator ϕ consists of learnable a Graph Attention Network (GAT) , which is used for separating graph into sub-graphs[35], and aggregator ψ play the role as aggregating sub-graphs to clustering labels according to sub-graphs.

The meta-training phase of LANDER was conducted on multi clustering tasks one by one. For example, we have n_{train} training tasks, and each task has a single cell transcriptional expression profile with manually annotated ground truth. At first, after pre-processed and de-noised, we obtained i^{th} reconstructed expression profile $\tilde{X}_i = h_i(X_i + \Delta_i)$ with Gaussian noise Δ_i. The ith auto-encoder h_i consists of encoder and decoder mentioned above, and then we built K-nearest neighbor (KNN) with DGL toolkit to build graph for every reconstructed expression $G_i^{(0)} = \{V_i^{(0)}, E_i^{(0)}\}$ where origin nodes $V_i^{(0)}$ are cells whose node features $X_i^{(0)}$ are reconstructed expression, i.e. $X_i^{(0)} = \tilde{X}_i$, and edges $E_i^{(0)}$ are determined by k nearest neighbors[36]. Noticeably, LANDER is dismountable and was pre-trained independently from rest parts of MeHi-SCC framework.

As a hierarchical clustering method, multi shared-parameters LANDERs are applied to divide current KNN graphs into sub-graphs layer by layer. Here we define $G_i^{(j)}$ as the graph of the ith training task/loop and the output of jth LANDER. The jth LANDER (also called "j^{th} level") has its separator $\phi^{(j)}$. Edges are subset of the input graph edges indicating that $E_i^{(j)} \subset E_i^{(j-1)}$ and are calculated via the separator that $E_i^{(j)} = \phi^{(j)}(G_i^{(j-1)}, X_i^{(j-1)})$. The new graph will be constructed based on split edge components $G_i^{(j)} = \{V_i^{(j-1)}, E_i^{(j)}\}$. Then aggregator ψ will be utilized to calculate node features of the next level, i.e. $X_i^{(j+1)} = \psi(X_i^{(j)}, G_i^{(j)})$. Here the aggregator compress every connected nodes pair into a single node for the next level. The construction of new edge set $E_i^{(j+1)}$ is still based on KNN searching on $X_i^{(j+1)}$. When no more edges are added, i.e. $E_i^{(j)} = \emptyset$, the split reaches convergence (i.e. early stopping). Finally LANDER will conclude and assign every cell with a clustering label according to which sub-graph it belongs to from the input level to the convergence level.

To be more precise, LANDER conducts clustering job without reliance on specific features, but on relationship between features and their neighborhood edges, i.e. the graph. Thus graph densities and linkages matter. For each edge we calculate density via two vertices $(v_m, v_n) \in E$ and linkage probabilities $p_{mn} = P(y_m = y_n)$ by a Multi Layer Perceptron (MLP) network with Softmax as the output transferring function. Their concatenated feature $[x_m, x_n]$ of the vertices pair (v_m, v_n) is used as input of the MLP network. Here y_m is the manually annotated cell type labels of v_m given from the support datasets. The node density $\hat{d}_m$ of cell v_m was calculated within k nearest neighbors for every node as $\hat{d}_m = \frac{1}{k}\sum_{j=1}^{k} \hat{e}_{mn} \cdot a_{mn}$, where $\hat{e}_{mn} = P(y_m = y_n) - P(y_m \neq y_n)$ is the edge coefficient and a_{mn} is the inner product of node features between vertices pair (v_m, v_n), which is defined as $a_{mn} = \langle x_m, x_n \rangle$

Separator ϕ is used to update edge connections for LANDER output. With pre-given connection latch p_τ (tuned during meta-training phase and stay fixed after meta-training, and for test clustering label estimation, the parameter is usually set 0.1), potential cell set for v_m is defined as $\Xi_m = \{n | (v_m, v_n) \epsilon E, \hat{d}_m < \hat{d}_n, p_{mn} < p_\tau\}$, and final edges linked with the cells that $\{v_n | n = argmax\ \hat{e}_{mn}, n \in \Xi_m\}$ are under the precondition that $\Xi_m \neq \emptyset$.

After the jth separator $\phi^{(j)}$ separates a certain graph $G^{(j)}$ into P sub-parts $g_1^{(j)}, g_2^{(j)}, \cdots, g_c^{(j)}, \cdots, g_P^{(j)}$, $g_c^{(j)} \subset G^{(j)}, c = 1,2,\cdots,P$, the cth sub-part $g_c^{(j)}$ will be first converted to a single node $v_c^{(j+1)}$ for the next level ($G^{(j+1)} = \{V^{(j+1)}, E^{(j+1)}\}, v_c^{(j+1)} \in V^{(j+1)}$), whose feature consists of two concatenated parts $x_c^{(j+1)} = [\hat{x}_c^{(j+1)}, \bar{x}_c^{(j+1)}]$, $x_c^{(j+1)} \subset X_i^{(j+1)}$. Here $\hat{x}_c^{(j+1)} = \hat{x}_\sigma^{(j)}, \sigma = \underset{n \in g_c^{(j)}}{\operatorname{argmax}}\ \hat{d}_n^{(j)}$ is the source feature who represents the peak node feature (the node with highest density) from the related sub-graphs, and $\bar{x}_c^{(j+1)} = \frac{1}{P}\sum_{n \in g_c^{(j)}} \hat{x}_n^{(j)}$ is the destination feature which denotes average feature of the related sub-graph ($\hat{x}_c^{(0)} = \bar{x}_c^{(0)} = x_c, c = 1,2,\cdots,N, x_c \subset \tilde{X}_i$):

Here, suppose d_m as cell density calculated by ground truth cell type label and ground-truth-based edge coefficients are calculate by $e_{mn} = I(y_m = y_n) - I(y_m \neq y_n)$ where $I(\cdot)$ is the indicator function. The two optimization goals in the meta-training phase are on nodes and edges respectively. The first is the density loss function defined as below to approximate estimated cell densities to ground truth densities:

$$L_d = \frac{1}{N}\sum_{m=1}^{N} \left\| d_m - \hat{d}_m \right\|_2^2 \tag{1}$$

Another optimization goal has restriction on edges, i.e. average edge cross-entropy to form nodes on each side of an edge linkage to a same cluster:

$$L_e = \frac{1}{|E|}\sum_{(v_m, v_n) \in E} l_{mn} \tag{2}$$

$$l_{mn} = \begin{cases} -q_{mn}\log p_{mn} - (1 - q_{mn})\log(1 - p_{mn}) & , if\ d_m < d_n \\ 0 & , else \end{cases} \tag{3}$$

Then the total loss of LANDER meta training is:

$$L_{meta} = L_d + L_e \tag{4}$$

C. Dual Self-Supervision for Further Clustering

Finally, after LANDERs are pre-trained with meta-training phase, we then apply residual graph convolution to further optimize clustering results. We constructed graph from dot-product similarity matrix $S \in \mathbb{R}^{N \times N}$ with de-noising encoder embedding $Z_{AE} = h_e(X)$, $Z_{AE} = \{z_1, z_2, \cdots, z_m, \cdots, z_n, \cdots, z_N\}^T$, where $z_m \in \mathbb{R}^z, m = 1,2,\cdots,N$. We calculate similarity between two cells, $s_{mn} = \frac{z_m z_n}{|z_m||z_n|}$, $s_{mn} \in S$:

Here we use similarity matrix to select k' ($k' = \min\{0.01N, 20\}$) cells as neighbors to construct graph adjacent matrix A. The residual graph is introduced in to make up for structure information that the de-noising auto-encoder missed:

$$H^{(k+1)} = \sigma\left((1-\alpha_k)\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(k)} + \alpha_k\tilde{X}\right)\left((1-\beta_k)I_N + \beta_k W^{(k)}\right) \quad (5)$$

Where σ depicts activation function RELU, parameters are $\alpha_k = 0.3$ and $\beta_k = 0.5 \times (k')^{-1}$ according to reference[37]. Normalized degree matrix $\tilde{D}$ is calculated via adjacent matrix with self-loop $\tilde{D} = \Sigma_j \tilde{A}_{ij}$ and $\tilde{A} = A + I_N$.

We then obtained final latent space of the residual graph network as a confidence matrix $Z_g = softmax(W^{(k)}H^{(k)} + b^{(k)})$, $Z_g \in \mathbb{R}^{N\times c}$ and c is the number of cell types given by hierarchical LANDER.

From pre-proposed research, Student's t-distribution helps optimize clustering. We fetched clustering labels given by LANDER as Y_0 and calculated the lth clustering centers μ_l for each cluster as $\mu_l = \frac{1}{n}\sum_i \tilde{z}_i$.

Where $\tilde{z}_i \subset \tilde{Z}$ is latent cell space whose clustering label is y_l and $y_l \in Y_0$. Hereby we are able to calculate similarity between the mth cell and l^{th} clustering center as q_{ml}, which can be regarded as the confidence degree (or the possibility) that the mth cell belongs to the lth cluster:

$$q_{ml} = \frac{(1+\|z_m-\mu_l\|^2/\upsilon)^{-\frac{\upsilon+1}{2}}}{\Sigma_{l'}(1+\|z_m-\mu_{l'}\|^2/\upsilon)^{-\frac{\upsilon+1}{2}}} \quad (6)$$

Here υ denotes the freedom degree and is set 1 by default. And then the assistant distribution is introduced in for promotion:

$$p_{ml} = \frac{q_{ml}^2/\Sigma_m q_{ml}}{\Sigma_{l'}(q_{ml'}^2/\Sigma_m q_{ml'})} \quad (7)$$

In all, supposed $P = [p_{ml}]$ and, the gold of minimizing the difference between the soft distribution P and the confidence degree matrix Q via KL divergence (Formula (8)) helps separation of representative learning. The similarity between soft distribution P to graph-network-learnt embedding Z_g acts likewise (Formula (9)). Thus the final training loss function is be defined as (10):

$$L_{pq} = KL(P||Q) = \Sigma_m\Sigma_l p_{ml}\log\frac{p_{ml}}{q_{ml}} \quad (8)$$

$$L_{gcn} = KL(P||Z_g) = \Sigma_m\Sigma_l p_{ml}\log\frac{p_{ml}}{z_{g,ml}} \quad (9)$$

$$L = MSE(X,\tilde{X}) + \theta L_{pq} + \eta L_{gcn} \quad (10)$$

Here, $z_{g,ml} \subset Z_g$, and θ is set as 0.1 and η as 0.01 for all query test datasets. After 1000 epochs of joint training with (10), the final general clustering result Y' of cell m will be calculated as $y'_m = \underset{l}{\operatorname{argmax}}\, z_{g,ml}$.

Noticeably, Y' shows further optimized aggregated clustering labels from LANDER clusters, while Y'' generated from formula (7) manifests more detailed sub-clusters within Y'. Y'', whose components y''_m are calculated as follows ($p_{g,ml}$ have similar definition with $z_{g,ml}$ and are components of P), is regarded as detailed sub-cluster result as $y''_m = \underset{l}{\operatorname{argmax}}\, p_{g,ml}$.

III. RESULTS

A. Performance on 14-1 Meta Training

Here we evaluate performance between benchmark methods on 13 normal scale datasets, and MeHi-SCC shows great superiority. Before dual self-supervision module, we trained LANDER for every query task with other 14 datasets' clustering tasks as support tasks within all our 15 datasets. The meta-training phase bases on multi-Ks (i.e.[4, 5, 38]) for DGL graph construction and multi-levels (i.e.[1, 3, 4]). The meta-training's learning rate is set 0.01 with stochastic gradient descent optimizer and batch size is set 4096 for 1000 epochs for each support tasks[39].

After LANDER's training was done, we applied several Ks (ranging from 3 to 10) and levels (ranging from 2 to 5) to test their potential highest performance. In terms of the dual self-supervision module, we first applied LANDER on reconstructed expression profiles after 400 epochs of de-noising pre-training to fetch the LANDER clustering labels and related clustering centers. The 400 epochs pre-training was based on Adam optimizer[40].

TABLE I. AVERAGE PERFORMANCE COMPARISON ON 13 NORMAL SCALE DATASETS

Methods	ARI	NMI	CA
CIDR[28]	0.4955	0.6476	0.6195
DCA[9]	0.5035	0.6306	0.5989
IDEC[22]	0.4708	0.5888	0.5952
scDeepCluster[20]	0.5600	0.7299	0.6433
Seurat[26]	0.5792	0.7474	0.6765
SIMLR[27]	0.5368	0.6721	0.6518
scGAC[24]	0.6319	0.7268	0.7277
scNAME[25]	0.6447	0.7192	0.7476
MeHi-SCC without meta-learning	0.6031	0.6820	0.7148
MeHi-SCC	**0.7698**	**0.7752**	**0.8119**

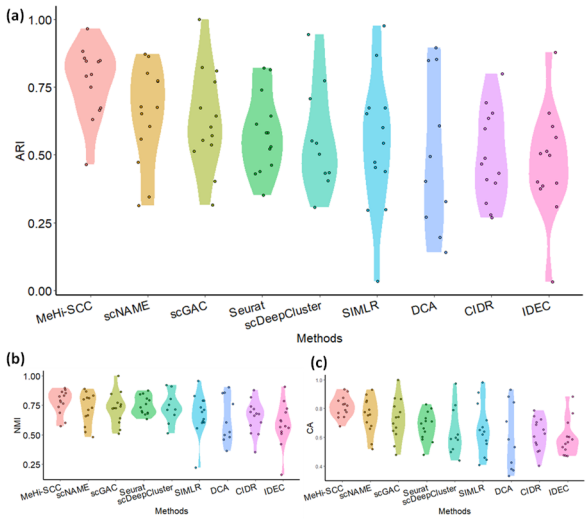

Fig. 2. Overall Performance on 13 Normal Scale Datasets. (a) ARI , (b) NMI , (c) CA.

We mainly use three indices for evaluation, i.e. Adjusted Rand Index (ARI), Normalized Mutual Information (NMI) and Clustering Accuracy (CA), calculated by sklearn package[41] (Supplementary Materials Formula S1-S3) .

In order to prove the effectiveness of meta-learning, we removed meta-training phase and only preserved hierarchical clustering in our framework with other conditions fixed ("MeHi-SCC without meta-learning" in Table I.), and the performance sees significant decrease without meta-learning.

Generally, MeHi-SCC outperformed all its counterpart methods(Table I.). In Fig.2, we plot all ARI, NMI and CA

performance with violin plots. On each plot, each violin bar includes 13 points indicating 13 normal scale datasets. Finally, MeHi-SCC obtained 0.7698 ARI on average (Fig.2(a)), exceeding scNAME (0.6447) by 19.4%, and scGAC (0.6319) by 21.8%. The average NMI were 0.7752 (Fig.2(b)), outperformed Seurat (0.7474) by 3.7% and scDeepCluster (0.7299) by 6.2%. The average CA finally reached 0.8119 on average(Fig.2(c)), which is 8.6% higher than scNAME (0.7476) and 11.6 % higher than scGAC (0.7277).

B. *Ablation Experiment*

TABLE II. ABLATION EXPERIMENTS ON DIFFERENT CLUSTERING CENTER TYPES

Clustering Center Type	ARI	NMI	CA
LANDER Whole Feature + PCA	0.3412	0.5610	0.4533
LANDER Src./Dst. Feature + PCA	0.2266	0.4503	0.3568
LANDER Whole Feature + AE	0.3320	0.5459	0.4448
LANDER Clusters Average	**0.7815**	**0.7827**	**0.8150**
K-means	0.6776	0.7419	0.7316

The connection between LANDER and the following dual self-supervision module plays an important role in overall performance. Thus we examined influence of different kinds of clustering centers towards overall performance. The five types of clustering centers includes:

(1) LANDER Whole Feature + PCA: We reduced dimension of the last layer (level) of LANDER graph features to the same dimension as the latent auto-encoder representations with principle component analysis (PCA)[42].

(2) LANDER Src./Dst. + PCA: Since the source feature and the destination feature, are the same in the last level, we fetched them for PCA dimension reduction. Different from LANDER Whole Feature, the source/destination feature only has half of the LANDER whole feature dimension.

(3) LANDER Whole Feature + AE. Here we reduced LANDER whole feature dimension with the de-noising auto-encoder we pre-trained instead of PCA to the latent shape as clustering centers.

(4) LANDER Clusters Average. This usage is what MeHi-SCC takes. Different from other types of centers mentioned about, here we do not use LANDER features as clustering centers but only use LANDER clustering results whose geometric centers are considered as clustering centers.

(5) K-means. The clustering centers of K-means are actually geometric centers of each clusters calculated by K-means. For fair comparison, we set K in K-means the same to the number of clusters calculated by LANDER.

All the ablation experiments were conveyed on same conditions, and then we measured average performance for 13 normal scale datasets (Table II). From UMAP visualization (Supplementary Materials Fig.S1), we spot that graph features generated by LANDER, with lower variance and regardless of dimension reduction methods, hardly fit in with other cells so that they cannot be used as clustering centers(red spots). In contrast, both K-means centers and LANDER clusters average centers performed better in that their clustering centers evenly allocated inside the entire cell manifolds. Thanks to the merged information from meta-learning, LANDER clusters average centers exceeded K-means by 15.3%, 5.5% and 11.4% on ARI, NMI and CA respectively.

C. *Visualization for Clustering Latent Spaces*

Then we fetched embedding representation of three representative benchmark pipelines, scNAME, scGAC and Seurat, together with MeHi-SCC for visualization with ground truth. Latent space from MeHi-SCC learnt a good representation with clear separation between basic cell types (Fig.3). We retrieved latent embedding spaces from MeHi-SCC de-noising auto-encoder after the whole training phase. For the rest of the benchmarks, we utilized default parameters. All the methods was visualized by UMAP. For those benchmarks based on R, we applied package umap to calculate dimension reduced coordinates, and for benchmarks based on python sessions, we visualized their latent space via packages scanpy and anndata[43].

MeHi-SCC sees clear separation on Baron Human dataset, compared with other benchmark methods. In scNAME latent space, Beta cells are mixed with mast cells, and activated stellate cells are mixed with quiescent stellate cells. The later phenomenon is also shown in scGAC latent space. In terms of Seurat, it generally performed well except that T cells are blended with Beta cells. Additionally, when it comes to Tasic, a small scale dataset, most of the benchmark pipelines

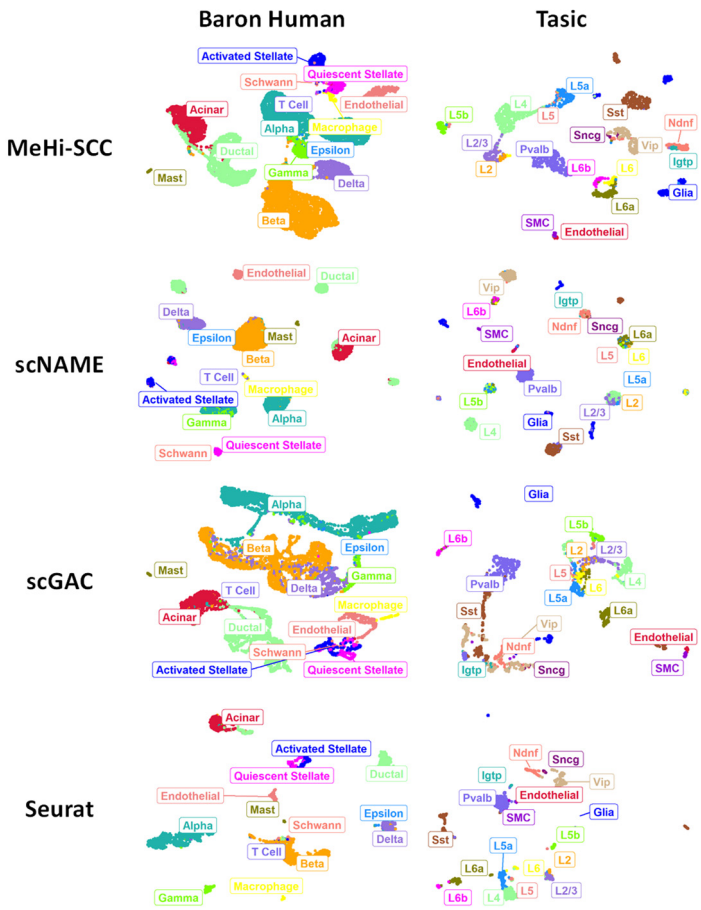

Fig. 3. Separation Performance Comparison on Latent Space of MeHi-SCC and Three Representative Methods on two example datasets with UMAP.

separate cells well, except that, for scGAC, it is difficult to separate L2, L2/3, L5 and L5a cells.

D. Case Studies of Sub-Cell Types Detection

Since MeHi-SCC is a hierarchical method, here we give two case studies of successful subtype detection. We applied gene enrichment analysis based on Seurat v3[26], then we retrieved detailed clustering results generated by MeHi-SCC on two large scale dataset PBMC 68K and Shekhar Mouse Retina.

After log-normalization and 2000 high variable genes selected, we then scaled the pre-normalized expression profiles, and ran PCA analysis to find neighbors with top 10 sPCs. Then we ran FindAllMarkers function in Seurat with min.pct=0.25 and filtered genes with log fold change threshold >1 and P value under 0.01 in each cluster. We here visualized the two datasets with latent spaces from MeHi-SCC, with ground truth labels (Fig.4(a), Supplementary Materials Fig.S2(a)) and detailed MeHi-SCC clustering results(Fig.4(b), Supplementary Materials Fig.S2(b)). After low quality clusters were filtered, we then referred to CellMarker database to validate potential clusters as cell sub-types[44].

In PBMC 68K, We found that CD37 enriched in cluster 10 and cluster 1[45], and MS4A1 enriched in cluster 10 indicating that cluster 10 could be identified as memory B cells among the CD19+ mature B cells[46]. GZMK is enriched in cluster 2 and cluster 23 indicating that we can identified effector memory CD8+ T cell inside CD8+ cytotoxic T cells[47]. Additionally, we found MZB1 and TNFRSF17 are enriched in cluster 20, seeing that MeHi-SCC deciphered the isolated part of the CD19+ B cells as dividing plasma B cells (Fig. 4(c))[45, 48, 49]. In term of Shekhar Mouse Retina, we found that Nfia is highly expressed especially in cluster 28 which is a potential mark for Rheaume et, al. 28 cells. Likewise, Gjc1 enriched in cluster 27 depicting markers of Rheaume et, al. 11 cells(Supplementary Materials Fig.S2(c))[50].

We then ran Wilcoxon signed rank test on marker genes detected between MeHi-SCC clusters mentioned above and the rest part of cells who belong to the same ground truth cell types with the clusters above. We spot that all potential genes expressed are significantly expressed, with P-value under 0.01(Fig.4(d), Supplementary Materials Fig.S2(d)).

IV. Discussion and Conclusion

In summary, we proposed MeHi-SCC, a meta-learning based graph hierarchical clustering framework for scRNA-seq data. In this work, we bring in extra information from multi-source scRNA datasets to aid clustering. MeHi-SCC outperformed current prevailing single cell clustering methods and the hierarchical clustering mechanism helps decipher sub-cell-types within known cell types in large-scale cell atlas.

Additionally, we encourage following researchers to test whether meta-learning clustering can be applied to more sparse single cell omics data like scATAC data. With more information provided by whole genomic sequencing, scRNA-seq could be an ideal assistant in referring clusters for other omics. And whether we can use multi bulk transcriptional profiles as assistance for single cell analysis with aid of meta-learning deserves exploring.

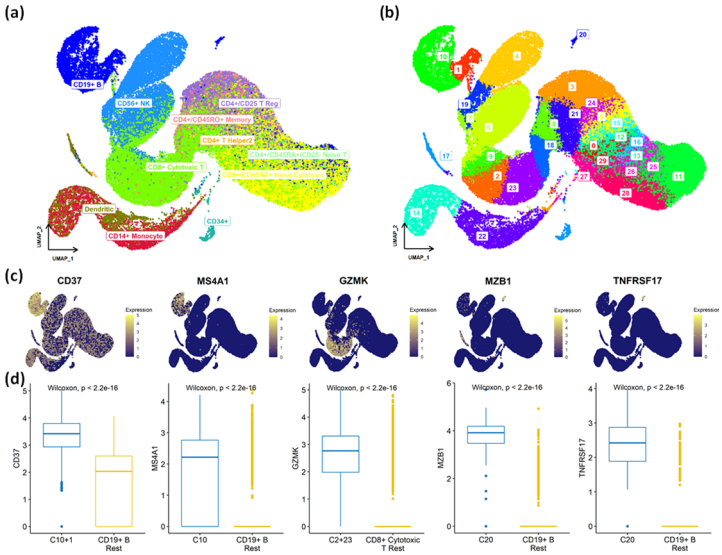

Fig. 4. Cell Subtype Detection on PBMC 68K by MeHi-SCC. (a) Ground Truth Cell Types of PBMC 68K, (b) Clustering Results Given by MeHi-SCC on PBMC 68K, (c) Five Potential Marker Genes Visualized on PBMC 68K, (d) Significance Tests for CD37. MS4A1. GZMK. MZB1 and TNFRSF17.

Acknowledgment

This work was sponsored by National Natural Science Foundation of China (61772566) and Guangdong Key Field R&D Plan (2019B020228001).

Supplementary Information

Codes, example data and supplementary materials are available at *https://github.com/biomed-AI/MeHi-SCC*.

References

[1] L. Yan *et al.*, "Single-cell RNA-Seq profiling of human preimplantation embryos and embryonic stem cells," *Nature Structural & Molecular Biology,* vol. 20, no. 9, pp. 1131-1139, 2013/09/01 2013.

[2] F. Tang, "Small RNAs in mammalian germline: Tiny for immortal," *Differentiation,* vol. 79, no. 3, pp. 141-146, 2010/03/01/ 2010.

[3] Evan Z. Macosko *et al.*, "Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets," *Cell,* vol. 161, no. 5, pp. 1202-1214, 2015/05/21/ 2015.

[4] S. J. Clark *et al.*, "scNMT-seq enables joint profiling of chromatin accessibility DNA methylation and transcription in single cells," *Nature Communications,* vol. 9, no. 1, p. 781, 2018/02/22 2018.

[5] J. Cao *et al.*, "Comprehensive single-cell transcriptional profiling of a multicellular organism," *Science,* vol. 357, no. 6352, pp. 661-667, 2017/08/18 2017.

[6] L. Hood and L. Rowen, "The Human Genome Project: big science transforms biology and medicine," *Genome Medicine,* vol. 5, no. 9, p. 79, 2013/09/13 2013.

[7] D. R. Glass *et al.*, "An Integrated Multi-omic Single-Cell Atlas of Human B Cell Identity," *Immunity,* vol. 53, no. 1, pp. 217-232.e5, 2020/07/14/ 2020.

[8] M. Andreatta, J. Corria-Osorio, S. Müller, R. Cubas, G. Coukos, and S. J. Carmona, "Interpretation of T cell states from single-cell transcriptomics data using reference atlases," *Nature Communications,* vol. 12, no. 1, p. 2965, 2021/05/20 2021.

[9] G. Eraslan, L. M. Simon, M. Mircea, N. S. Mueller, and F. J. Theis, "Single-cell RNA-seq denoising using a deep count autoencoder," *Nature Communications,* vol. 10, no. 1, p. 390, 2019/01/23 2019.

[10] H. T. N. Tran *et al.*, "A benchmark of batch-effect correction methods for single-cell RNA sequencing data," *Genome Biology,* vol. 21, no. 1, p. 12, 2020/01/16 2020.

[11] P. He *et al.*, "The changing mouse embryo transcriptome at whole tissue and single-cell resolution," *Nature,* vol. 583, no. 7818, pp. 760-767, 2020/07/01 2020.

[12] J. Han, R. A. DePinho, and A. Maitra, "Single-cell RNA sequencing in pancreatic cancer," *Nature Reviews Gastroenterology & Hepatology,* vol. 18, no. 7, pp. 451-452, 2021/07/01 2021.

[13] M. W. Berry, "Large-Scale Sparse Singular Value Computations," *The International Journal of Supercomputing Applications,* vol. 6, no. 1, pp. 13-49, 1992/04/01 1992.

[14] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "UMAP: Uniform Manifold Approximation and Projection," *Journal of Open Source Software,* vol. 3, p. 861, 09/02 2018.

[15] L. van der Maaten and G. Hinton, "Viualizing data using t-SNE," *Journal of Machine Learning Research,* vol. 9, pp. 2579-2605, 11/01 2008.

[16] J. Rao, X. Zhou, Y. Lu, H. Zhao, and Y. Yang, "Imputing single-cell RNA-seq data by combining graph convolution and autoencoder neural networks," *iScience,* vol. 24, no. 5, 2021.

[17] Y. Zeng, Z. Wei, F. Zhong, Z. Pan, Y. Lu, and Y. Yang, "A parameter-free deep embedded clustering method for single-cell RNA-seq data," *Brief Bioinform,* vol. 23, no. 5, Sep 20 2022.

[18] Y. Zeng, Z. Wei, Z. Pan, Y. Lu, and Y. Yang, "A robust and scalable graph neural network for accurate single-cell classification," *Briefings in Bioinformatics,* vol. 23, no. 2, 2022.

[19] X. Zhou, H. Chai, Y. Zeng, H. Zhao, and Y. Yang, "scAdapt: virtual adversarial domain adaptation network for single cell RNA-seq data classification across platforms and species," *Brief Bioinform,* vol. 22, no. 6, Nov 5 2021.

[20] T. Tian, J. Wan, Q. Song, and Z. Wei, "Clustering single-cell RNA-seq data with a model-based deep learning approach," *Nature Machine Intelligence,* vol. 1, no. 4, pp. 191-198, 2019/04/01 2019.

[21] X. Junyuan, G. Ross, and F. Ali, "Unsupervised Deep Embedding for Clustering Analysis," 2016/06/11

[22] X. Guo, L. Gao, X. Liu, and J. Yin, *Improved Deep Embedded Clustering with Local Structure Preservation*. 2017.

[23] Y. Zeng, X. Zhou, J. Rao, Y. Lu, and Y. Yang, "Accurately Clustering Single-cell RNA-seq data by Capturing Structural Relations between Cells through Graph Convolutional Network," in *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020, pp. 519-522.

[24] Y. Cheng and X. Ma, "scGAC: a graph attentional architecture for clustering single-cell RNA-seq data," *Bioinformatics,* vol. 38, no. 8, pp. 2187-2193, 2022.

[25] H. Wan, L. Chen, and M. Deng, "scNAME: neighborhood contrastive clustering with ancillary mask estimation for scRNA-seq data," *Bioinformatics,* vol. 38, no. 6, pp. 1575-1583, 2022.

[26] T. Stuart *et al.*, "Comprehensive Integration of Single-Cell Data," *Cell,* vol. 177, no. 7, pp. 1888-1902.e21, 2019.

[27] B. Wang, D. Ramazzotti, L. De Sano, J. Zhu, E. Pierson, and S. Batzoglou, "SIMLR: A Tool for Large-Scale Genomic Analyses by Multi-Kernel Learning," *PROTEOMICS,* vol. 18, no. 2, p. 1700232, 2018/01/01 2018.

[28] P. Lin, M. Troup, and J. W. K. Ho, "CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data," *Genome Biology,* vol. 18, no. 1, p. 59, 2017/03/28 2017.

[29] X. Li, Z. Sun, J.-H. Xue, and Z. Ma, "A concise review of recent few-shot meta-learning methods," *Neurocomputing,* vol. 456, pp. 463-468, 2021/10/07/ 2021.

[30] H.-y. Lee, S.-W. Li, and T. Vu, *Meta Learning for Natural Language Processing: A Survey*. 2022, pp. 666-684.

[31] S. Luo, Y. Li, P. Gao, Y. Wang, and S. Serikawa, "Meta-seg: A survey of meta-learning for image segmentation," *Pattern Recognition,* vol. 126, p. 108586, 2022/06/01/ 2022.

[32] Y. Xing *et al.*, *Learning Hierarchical Graph Neural Networks for Image Clustering*. 2021.

[33] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems,* vol. 32, 2019.

[34] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics,* vol. 22, no. 1, pp. 79-86, 1951.

[35] Z. Liu and J. Zhou, "Graph Attention Networks," 2020, pp. 39-41.

[36] M. Wang *et al.*, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," *arXiv preprint arXiv:1909.01315,* 2019.

[37] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li, "Simple and Deep Graph Convolutional Networks," presented at the Proceedings of the 37th International Conference on Machine Learning, Proceedings of Machine Learning Research, 2020.

[38] U. C. Eze, A. Bhaduri, M. Haeussler, T. J. Nowakowski, and A. R. Kriegstein, "Single-cell atlas of early human brain development highlights heterogeneity of human neuroepithelial cells and early radial glia," *Nature Neuroscience,* vol. 24, no. 4, pp. 584-594, 2021/04/01 2021.

[39] D. Saad, "Online algorithms and stochastic approximations," *Online Learning,* vol. 5, no. 3, p. 6, 1998.

[40] D. P. Kingma and L. J. Ba, "Adam: A Method for Stochastic Optimization," presented at the International Conference on Learning Representations (ICLR), 2015, 2015.

[41] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research,* vol. 12, pp. 2825-2830, 2011.

[42] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 2, no. 4, pp. 433-459, 2010/07/01 2010.

[43] F. A. Wolf, P. Angerer, and F. J. Theis, "SCANPY: large-scale single-cell gene expression data analysis," *Genome Biology,* vol. 19, no. 1, p. 15, 2018/02/06 2018.

[44] X. Zhang *et al.*, "CellMarker: a manually curated resource of cell markers in human and mouse," (in eng), *Nucleic Acids Res,* vol. 47, no. D1, pp. D721-d728, Jan 8 2019.

[45] J.-Y. Zhang *et al.*, "Single-cell landscape of immunological responses in patients with COVID-19," *Nature Immunology,* vol. 21, no. 9, pp. 1107-1118, 2020/09/01 2020.

[46] F. Qi, W. Zhang, J. Huang, L. Fu, and J. Zhao, "Single-Cell RNA Sequencing Analysis of the Immunometabolic Rewiring and Immunopathogenesis of Coronavirus Disease 2019," (in English), *Frontiers in Immunology,* Original Research vol. 12, 2021-April-14 2021.

[47] F. Zhang *et al.*, "Adaptive immune responses to SARS-CoV-2 infection in severe versus mild individuals," *Signal Transduction and Targeted Therapy,* vol. 5, no. 1, p. 156, 2020/08/14 2020.

[48] Z. Huang *et al.*, "Effects of sex and aging on the immune cell landscape as assessed by single-cell transcriptomic analysis," *Proceedings of the National Academy of Sciences,* vol. 118, no. 33, p. e2023216118, 2021/08/17 2021.

[49] J. P. Bernardes *et al.*, "Longitudinal Multi-omics Analyses Identify Responses of Megakaryocytes, Erythroid Cells, and Plasmablasts as Hallmarks of Severe COVID-19," *Immunity,* vol. 53, no. 6, pp. 1296-1314.e9, 2020/12/15/ 2020.

[50] B. A. Rheaume *et al.*, "Single cell transcriptome profiling of retinal ganglion cells identifies cellular subtypes," *Nature Communications,* vol. 9, no. 1, p. 2759, 2018/07/17 2018.