Exploring the Interaction between Local and Global Latent Configurations for Clustering Single-Cell RNA-Seq: A Unified Perspective

Nairouz Mrabah, Mohamed Mahmoud Amar, Mohamed Bouguessa, Abdoulaye Banire Diallo

University of Quebec at Montreal, Montreal, Quebec, Canada
{mrabah.nairouz, amar.mohamed_mahmoud}@courrier.uqam.ca, {bouguessa.mohamed, diallo.abdoulaye}@uqam.ca

Abstract

The most recent approaches for clustering single-cell RNA-sequencing data rely on deep auto-encoders. However, three major challenges remain unaddressed. First, current models overlook the impact of the cumulative errors induced by the pseudo-supervised embedding clustering task (Feature Randomness). Second, existing methods neglect the effect of the strong competition between embedding clustering and reconstruction (Feature Drift). Third, the previous deep clustering models regularly fail to consider the topological information of the latent data, even though the local and global latent configurations can bring complementary views to the clustering task. To address these challenges, we propose a novel approach that explores the interaction between local and global latent configurations to progressively adjust the reconstruction and embedding clustering tasks. We elaborate a topological and probabilistic filter to mitigate Feature Randomness and a cell-cell graph structure and content correction mechanism to counteract Feature Drift. The Zero-Inflated Negative Binomial model is also integrated to capture the characteristics of gene expression profiles. We conduct detailed experiments on real-world datasets from multiple representative genome sequencing platforms. Our approach outperforms the state-of-the-art clustering methods in various evaluation metrics.

Introduction

Single-cell RNA-sequencing (scRNA-seq) is a broadly adopted technique for measuring transcriptome-wide gene expression levels at the mono-cell resolution. The derived datasets can provide valuable insights into critical biological topics, such as developing a deeper understanding of complex tissues (Welch et al. 2019), the discovery of rare/novel cell types (Jindal et al. 2018), the study of complex diseases such as the detection of cancer stem cells (Bhaduri et al. 2020; Paik et al. 2020), the elucidation of cell heterogeneity (Papalexi and Satija 2018), the exploration of complex systems (Stubbington et al. 2017) and the inference of cell trajectory (Saelens et al. 2019). Despite being systematic, automatic cell annotations regularly fail to achieve accurate labeling and require a relevant reference dataset. Furthermore, expert-managed cell annotations are slow, expensive, and can be subjective. In this context, we investigate the unsupervised identification of cell types for scRNA-seq datasets.

As an unsupervised learning strategy, clustering constitutes the most effective method to accurately identify cell types (Kiselev, Andrews, and Hemberg 2019). It is a crucial step in almost any scRNA-seq data analysis. After the breakthrough of high-dimensional gene expression profiles, researchers have tried to address the clustering problem by combining dimensionality reduction techniques, such as PCA, T-SNE and UMAP (McInnes, Healy, and Melville 2018), with classical clustering methods, including K-means, hierarchical clustering (Johnson 1967), and the density-based methods (Ester et al. 1996). Unfortunately, due to the non-trivial relationships between within-cluster cells, the severe sparsity induced by dropout events, as well as the high variability of gene expression levels (Kiselev, Andrews, and Hemberg 2019), these approaches achieve suboptimal clustering results.

To overcome the limitations of the classical clustering methods, more advanced techniques have been introduced. Several methods, including scImpute (Li and Li 2018), CIDR (Lin, Troup, and Ho 2017), MAGIC (van Dijk et al. 2017), and SAVER (Huang et al. 2018), first impute the dropouts in the gene expression count matrix and then cluster the clean data. Although these approaches manage to reduce the number of false-zero values, they generally fail to capture the complex structure inherent to scRNA-seq data. To address this problem, kernel clustering methods, such as MPSSC (Park and Zhao 2018) and SIMLR (Wang et al. 2017) can learn robust similarity measures. However, these approaches can not match the expressive power of deep learning. Futhermore, the time/memory prohibitive costs can be a barrier to their use on the rapidly growing gene transcriptional datasets.

Recently, several auto-encoder-based models follow the deep clustering paradigm to identify cell clusters such as scDC (Tian et al. 2019), scDCC (Tian et al. 2021), and scziDesk (Chen et al. 2020). In addition, most of these approaches incorporate a zero-inflated negative bionomial (ZINB) model to characterize the discreteness, zero-inflation and over-dispersion of gene expression data. Although these deep models achieve considerable improvement in clustering performance, the structural relationships between cells are ignored, making the clustering task more difficult. This is why a new class of models (Liu et al. 2022) has emerged based on graph neural networks (GNNs), including scGNN (Wang et al. 2021), scGAE (Luo et al. 2021), scTAG (Yu et al. 2022) and scDSC (Gan et al. 2022).

In general, the above methods perform joint reconstruction (self-supervision) and embedding clustering (pseudo-supervision). Recently, Mrabah, Bouguessa, and Ksantini (2022) have shown that the relationship between self-supervision and pseudo-supervision is governed by the trade-off between Feature Randomness and Feature Drift. On the one hand, training with pseudo-labels constructed from the noisy clustering assignments leads to cumulative errors in the optimization process, which in turn corrupt the latent representations and reduce the clustering performance. This problem is called Feature Randomness (FR). On the other hand, the reconstruction task, which is performed to reduce FR, is in conflict with the goal of embedding clustering. In contrast to embedding clustering, the reconstruction is concerned with maintaining within- and between-cluster variances, rather than reducing within-cluster variance and increasing between-cluster variance. This problem is called Feature Drift (FD). The trade-off between FR and FD remains unaddressed by single-cell deep clustering methods.

We propose a novel single-cell deep graph clustering model that tackles the trade-off between FR and FD. Our method explores the interaction between local and global latent configurations to adjust the pseudo-supervised and self-supervised tasks. For the pseudo-supervised task, we propose a topological and probabilistic filter that mitigates the effect of FR. Our filter has two complementary operations. The first operation focuses on the global configuration, which is captured by a probabilistic model. The second filtering operation focuses on the local configuration by considering the topological representation of the data in the latent space. For the self-supervised task, we propose a progressive transformation strategy that counteracts the effect of FD. Our transformation relies on two complementary operations and gradually transforms the reconstruction into a clustering-oriented task. The first operation leverages the global configuration to generate a clustering-oriented gene expression matrix. The second operation handles the local configuration to generate a clustering-oriented graph structure.

Contributions. **(1)** We establish the first single-cell deep graph clustering model that tackles the trade-off between Feature Randomness and Feature Drift. **(2)** We propose a topological and probabilistic filter that mitigates the effect of Feature Randomness. **(3)** We devise a progressive cell-cell graph structure and content correction mechanism to counteract Feature Drift. **(4)** Our approach explores the interaction between global and local latent configurations to adjust the pseudo-supervised and self-supervised training tasks. Our local and global characterizations of the data in the latent space provide complementary information to the clustering task. **(5)** We carry out several experiments on representative real-world scRNA-seq datasets. The obtained results validate our contributions and show that the proposed method outperforms state-of-the-art single-cell clustering approaches.

Related Work

We divide the state-of-the-art methods for clustering scRNA-seq data into three main categories: deep embedding methods, deep clustering methods, and deep graph clustering methods. In Appendix A (all appendices are provided in the Supplementary Material[†]), we discuss additional related methods, which are not specific to the single-cell field of research.

The deep embedding methods perform clustering and embedding learning separately. In (Yu et al. 2021), scGMAI minimizes vanilla reconstruction using an auto-encoder network. After that, the fast independent component analysis (FastICA) algorithm is employed to reduce the reconstructed data dimension. The low-dimensional cell representations are finally clustered using Gaussian mixture models (GMMs). In (Eraslan et al. 2019), scDCA utilizes a denoising auto-encoder to reconstruct the gene expression count matrix. The decoding process is controlled by the ZINB distribution to capture the characteristics of the count matrix (discreteness, zero-inflation, and over-dispersion). However, both methods (i.e., scGMAI and scDCA) do not consider the clustering meta-prior to learn the latent configuration. As a result, scGMAI and scDCA fail to learn clustering-friendly latent features, which leads to suboptimal clustering results.

Incorporating the clustering inductive bias effectively requires joint pseudo-supervision and self-supervision. To overcome the limitations of scGMAI and scDCA, numerous single-cell clustering methods have adopted the deep clustering paradigm. For instance, scDC (Tian et al. 2019) improves the scDCA model. More precisely, it has added an embedding clustering task to the self-supervised denoising task using the Kullback-Leibler divergence between the clustering assignments distribution and a pseudo-supervised target distribution. In (Chen et al. 2020), scziDesk combines a soft k-means embedding clustering and a self-supervised denoising scheme. In (Tian et al. 2021), scDCC augments the deep clustering framework with domain specific knowledge by adding a constraint loss. However, scDCA, scziDesk, scDCC can not preserve the structural information between cells during the encoding process. Furthermore, these models overlook the trade-off between FR and FD. Last but not lest, scDCA, scziDesk, scDCC fail to consider the interaction between local and global latent configurations, which can provide complementary information to the clustering task.

Deep graph clustering methods supply the deep clustering framework with a message passing mechanism (Gilmer et al. 2017) to preserve the structural information during the encoding process. In (Luo et al. 2021), scGAE performs embedding clustering and reconstructs the gene count matrix and the cell graph adjacency matrix. However, the decoding process of this method ignores the characteristics of the gene count distribution. To reduce the detrimental effect of the prevalent false zeros, scGNN (Wang et al. 2021) performs several levels of reconstruction and aggregates cell–cell relationships using graph neural networks. This method relies on a left-truncated Gaussian mixture model to perform embedding clustering. However, scGNN falls short of a probabilistic process to generate the gene count data. For this reason, scTAG (Yu et al. 2022) embeds the ZINB distribution in a topology adaptive graph convolutional autoencoder that performs joint clustering and reconstruction. Compared to scTAG, scDSC (Gan et al. 2022) connects fully-connected encoding layers with a

[†]https://github.com/MMAMAR/scTPF

GNN module to maximize the mutual information extracted from both. However, all these models overlook the trade-off between FR and FD and fail to consider the interaction between local and global latent configurations.

Proposed Method

We consider the problem of clustering scRNA-seq data. Let $X = (x_{ij}) \in \mathbb{R}^{N \times M}$ be the gene expression count matrix, where N denotes the number of cells, M denotes the number of genes. Each element x_{ij} represents the expression count of the j^{th} gene in the i^{th} cell. The transcriptome of the i^{th} cell is denoted by x_i. We assume that the cells of X can be grouped into J clusters. The clustering indices are denoted by the vector $C = (c_i)_{1 \leqslant i \leqslant N}$, where $c_i \in \{1, ..., J\}$.

To handle the complex spread of the scRNA-seq data, we construct a cell graph to capture the structural information. Cells can be viewed as the vertices $\mathcal{V} = \{v_i\}_{i=1}^{N}$ of an undirected attributed graph $\mathcal{G}$. The structure of $\mathcal{G}$ is built based on the κ-NN algorithm. The number of neighbors for each node is denoted by κ. We use the Euclidean distance between cell transcriptomes to identify the nearest neighbors. An edge connects two nodes v_i and v_j if and only if x_i and x_j are neighbors within the κ shortest distance. The resulting graph adjacency matrix of $\mathcal{G}$ is denoted by A, where $a_{ij} = 1$ if v_i and v_j are connected, otherwise $a_{ij} = 0$. The node attributes of $\mathcal{G}$ are described by the gene expression count matrix X.

We devise a graph embedding strategy to preserve the structural information and extract high-level representations. More precisely, we use a graph auto-encoder (Kipf and Welling 2016). A graph convolutional encoder f_E with L_E layers projects the attributed graph $\mathcal{G}$ into a low-dimensional latent space. We denote by $Z \in \mathbb{R}^{N \times d}$ the output of the encoder, where d is the latent space dimension. The propagation rule of the l^{th} encoding layer is governed by the graph convolution operation (Kipf and Welling 2017) as described by:

$$Z^{(l)} = \text{ReLU}(\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} Z^{(l-1)} W_E^{(l)}), \tag{1}$$

where $Z^{(l)}$ and $W_E^{(l)}$ are, respectively, the output and training weights of the l^{th} encoding layer, such that $Z^{(0)} = X$ and $Z^{(L_E)} = f_E(X, A)$. The matrix $\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}$ denotes the normalized adjacency, where $\widetilde{A} = A + I$, $\widetilde{D} = \text{diag}(\widetilde{A}\, \mathbf{1}_N)$, $\mathbf{1}_N \in \mathbb{R}^N$ is a vector of ones, and $I \in \mathbb{R}^{N \times N}$ is the identity.

The decoder relies on two heads to generate a new attributed graph $\mathcal{G}^{gen}$. The first head constructs the structure $\hat{A} = (\hat{a}_{ij}) \in \mathbb{R}^{N \times N}$ of $\mathcal{G}^{gen}$ from the latent codes. The second head constructs the feature matrix $\hat{X} \in \mathbb{R}^{N \times M}$ using L_D fully-connected layers denoted by f_D. The layers of the second head have ReLU activations. We denote by $W_D^{(l)}$ the training weights of the l^{th} decoding layer. We perform two training phases to obtain the final clustering assignments.

First Phase: Pretraining

Before introducing the pseudo-supervised objective function (main task), we perform a pretraining phase based on self-supervision (pretext task). The pretext task learns high-level representations and precludes the extraction of pseudo-labels from the initially random latent codes.

We define a distribution $p(\hat{A}, \hat{X} \mid Z)$ that controls the decoding process. We factorize $p(\hat{A}, \hat{X} \mid Z)$ into two independent distributions $p(\hat{A} \mid Z)$ and $p(\hat{X} \mid Z)$ associated with the first and second decoding heads, respectively. The first distribution $p(\hat{A} \mid Z)$ captures the structure generation. We use a Bernoulli distribution to characterize the binary nature of $\hat{A}$. Accordingly, we formulate $p(\hat{A} \mid Z)$ as follows:

$$p(\hat{A}|Z) = \prod_{i,j}^{N} p(\hat{a}_{ij} \mid z_i,\, z_j) = \prod_{i,j=1}^{N} \mathcal{B}(\hat{a}_{ij} \mid \beta_{ij}), \tag{2}$$

where $p(\hat{a}_{ij} \mid z_i,\, z_j)$ is a Bernoulli distribution $\mathcal{B}\text{er}(\beta_{ij})$ parameterized by $\beta ij = \text{sigmoid}(z_i^T z_j)$.

The second distribution $p(\hat{X} \mid Z)$ captures the generation of node attributes. We use the ZINB distribution (Wang et al. 2021; Yu et al. 2022) to characterize the discreteness, zero-inflation (high-sparsity due to true and dropout zeros) and over-dispersion (variance greater than mean) of the scRNA-seq gene expression count matrix $\hat{X}$. Accordingly, we formulate $p(\hat{X} \mid Z)$ as described by:

$$p(\hat{X}|Z) = \prod_{i}^{N} p(\hat{x}_i \mid z_i) = \prod_{i}^{N} \text{ZINB}(\hat{x}_i \,|\pi_i,\, \mu_i,\, \sigma_i^2), \tag{3}$$

where $p(\hat{x}_i \mid z_i)$ is a ZINB distribution parameterized by three M dimensional vectors: the zero rate vector $\pi = \text{sigmoid}(W_\pi f_D(z_i))$, the mean vector of the associated negative binomial $\mu = \exp(W_\mu f_D(z_i))$, and the variance vector of the associated negative binomial $\sigma^2 = \exp(W_\sigma f_D(z_i))$; W_π, W_μ, W_σ are the weight matrices of π, μ, and σ^2, respectively. The derivation of the ZINB mass function based on these parameters (i.e., π, μ, σ^2) is discussed in Appendix B.

Our pretraining task consists of reconstructing the input graph $\mathcal{G}$. To this end, we minimize the negative log-likelihood of $p(\mathcal{G}^{gen} = \mathcal{G} \mid Z)$. The resulting self-supervised function L_{SS} can be formulated as follows:

$$L_{\text{SS}}(\mathcal{G}) = -\log(p(\mathcal{G}^{gen} = \mathcal{G} \mid Z)) = L_X + L_A, \tag{4}$$

where $L_A = -\log(p(\hat{A} = A \mid Z))$ is the reconstruction of the graph structure A, and $L_X = -\log(p(\hat{X} = X \mid Z))$ is the reconstruction of the gene expression count matrix X.

Second Phase: Clustering

Unlike the previous methods, which usually optimize a linear combination of embedding clustering and reconstruction, our strategy alleviates the trade-off between FR and FD. In Figure 1, we illustrate the framework of the clustering phase. Our approach has two components: i) a pseudo-supervised loss supplied with a topological and probabilistic filter $\mathcal{F}$ to gradually select the most reliable samples for embedding clustering, ii) a self-supervised loss supplied with a correction mechanism $\mathcal{T}$ to gradually transform the reconstruction into a clustering-oriented graph construction.

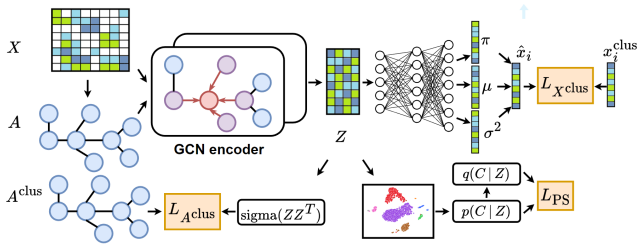

Figure 1: Illustration of the clustering phase of scTPF.

Pseudo-Supervision with a FR Protection Mechanism We define a clustering assignment distribution $p(C \mid Z)$ that captures the clustering structure. This distribution is factorized $p(C \mid Z) = \prod_{i=1}^{N} p(c_i \mid z_i)$. We denote by p_{ij} the probability $p(c_i = j \mid z_i)$, which is described by:

$$p_{ij} = \frac{(1 + \|z_i - \Omega_j\|^2)^{-1}}{\sum_{j'}(1 + \|z_i - \Omega_{j'}\|^2)^{-1}}, \tag{5}$$

where $\{\Omega_j\}_{j=1}^{J}$ is a set of trainable clustering centers. We initialize these centers by performing spectral clustering on A and projecting the obtained centers in the latent space.

Building accurate pseudo-labels from the noisy clustering assignments is a challenging task. Maennel et al. (2020) have shown that the impact of label randomness cannot be reversed by training a model on ground-truth labels after pretraining on random labels. Motivated by this intuition, we operate a protection mechanism rather than attempting to correct the noisy pseudo-labels after training with them. More precisely, we propose a filtering strategy that jointly exploits the local and global characteristics of the latent configuration. Our filter operates in two steps and aims to detect the samples with reliable pseudo-labels. As a result, the trained model progressively learns discriminative features and hence acquires the capacity to select more samples with reliable pseudo-labels.

The first filtering step $\mathcal{F}_1$ is a probability-based operation. We focus on the *global* configuration by considering the relationship between the latent codes and the latent centers. This relation is captured by the distribution $p(c_i \mid z_i)$. We denote by λ_i^1 (λ_i^2, respectively) the first (second, respectively) highest assignment score of $p(c_i \mid z_i)$. We filter out samples with a distance between λ_i^1 and λ_i^2 below a threshold ϵ.

The second filtering step $\mathcal{F}_2$ is a topology-based operation. We focus on the *local* configuration by considering the relationship between the latent codes and their k nearest neighbors. This relation is captured by the topology of an undirected graph $\mathcal{G}^{\text{latent}}$ built by applying the k-NN algorithm on the latent codes. We construct the subgraph of the i^{th} cluster $\mathcal{G}_i^{\text{latent}}$ by dropping the nodes of the other clusters and their corresponding edges. The cluster of each node is determined by the index j of $\max_j(p(c = j|z))$. Then, we identify the largest connected component Q_i of $\mathcal{G}_i^{\text{latent}}$. After that, we filter out isolated samples that do not belong to $\bigcup_i Q_i$.

We define Θ as the set of reliable samples for the embedding clustering task. We compute this set iteratively by performing the intersection between the outputs of the first and second filtering operations. The two complementary filtering operations explore local and global latent configurations to select the most relevant samples for embedding clustering. Our filter $\mathcal{F}$ is summarized in Algorithm 1.

Algorithm 1: Topological and Probabilistic Filter: $\mathcal{F}$

Input: latent codes: Z, clustering assignments: (p_{ij}), filtering threshold: ϵ, # of neighbors: k, # of clusters: J
Output: reliable samples: Θ

$\Theta_1, \Theta_2 \leftarrow \varnothing, \varnothing$;
for $i = 1$ **to** Z.shape[0] **do**
  $\lambda_i^1, \lambda_i^2 \leftarrow \max_j \{p_{ij}\}, \max_j \{p_{ij} \,|\, p_{ij} < \lambda_i^1\}$;
  $\Theta_1 \leftarrow \Theta_1 \cup \{i \in \mathcal{V} \mid |\lambda_i^1 - \lambda_i^2| \geqslant \epsilon\}$;
end for
Construct the graph $\mathcal{G}^{\text{latent}}$ over Z using k-NN;
for $i = 1$ **to** J **do**
  Construct $\mathcal{G}_i^{\text{latent}}$ with nodes only from the i^{th} cluster;
  Compute Q_i the largest connected comp. of $\mathcal{G}_i^{\text{latent}}$;
  $\Theta_2 \leftarrow \Theta_2 \cup Q_i$;
end for
Return $\Theta_1 \cap \Theta_2$;

We define a target clustering distribution $q(C \mid Z)$ that emphasizes the clustering structure using $\mathcal{F}$. This distribution is factorized $q(C \mid Z) = \prod_{i=1}^{N} p(c_i \mid z_i)$. We denote by q_{ij} the probability $q(c_i = j \mid z_i)$, which is described by:

$$q_{ij} = \begin{cases} 1 \text{ if } i \in \Theta \text{ and } j = \arg\max_{j'}(p_{ij'}), \\ 0 \text{ if } i \in \Theta \text{ and } j \neq \arg\max_{j'}(p_{ij'}), \\ p_{ij} \text{ if } i \notin \Theta. \end{cases} \tag{6}$$

The distribution $q(c_i \mid z_i)$ progressively constructs hard clustering assignments. Our pseudo-supervised loss is:

$$L_{\text{PS}} = \text{KL}\big(q(C|Z)||p(C|Z)\big) = \sum_{i=1}^{N}\sum_{j=1}^{J} q_{ij}\log(\frac{q_{ij}}{p_{ij}}), \tag{7}$$

where KL is the Kullback–Leibler divergence between the target distribution $q(C \mid Z)$ and the soft assignment distribution $p(C \mid Z)$. The samples filtered by $\mathcal{F}$ have a zero value for $\log\big(\frac{q_{ij}}{p_{ij}}\big)$ and are thus systematically discarded from L_{PS}.

Self-Supervision with a FD Correction Mechanism The clustering phase regularly involves a self-supervised pretext (reconstruction) to reduce FR, but leads to strong FD as a byproduct. A protection mechanism against FD consists of replacing the reconstruction with a clustering-oriented self-supervised task in a *onetime* transformation process. However, this strategy leads to excessive FR. Thus, we operate a correction mechanism instead of a protection one. More precisely, we propose a transformation strategy $\mathcal{T}$ that exploits the local and global characteristics of the latent data based on two complementary operations. Our transformation $\mathcal{T}$ progressively generates a more clustering-oriented graph $\mathcal{G}^{\text{clus}}$ from the initial graph $\mathcal{G}$. The structure and node content of $\mathcal{G}^{\text{clus}}$ are denoted by A^{clus} and X^{clus}, respectively.

Algorithm 2: Graph Progressive Transformation: $\mathcal{T}$

Input: latent codes: Z, clustering assignments: (p_{ij}), reliable samples: Θ, latent graph: $\mathcal{G}^{\text{latent}}$, initial graph: $\mathcal{G}$, clustering centers: $\{\Omega_j\}$

Output: clustering-oriented graph: $\mathcal{G}^{\text{clus}}$

Compute X^{clus} according to Eq. (8);
$A^{\text{clus}} \leftarrow A$;
/* indices($\mathcal{G}$, i) returns the nodes connected to the i^{th} node in $\mathcal{G}$ */
for i **in** Θ **do**
  for j **in** $\Theta \cap$ indices($\mathcal{G}^{\text{latent}}$, i) **do**
    $a_{ij}^{\text{clus}} \leftarrow 1$;
  end for
  for j **in** $\Theta \cap$ indices($\mathcal{G}$, i) **do**
    if $\arg\max_{j'}(p_{ij'}) \neq \arg\max_{j'}(p_{jj'})$ **then**
      $a_{ij}^{\text{clus}} \leftarrow 0$;
    end if
  end for
end for
Return $\mathcal{G}^{\text{clus}}$;

The first operation $\mathcal{T}_1$ gradually constructs X^{clus} from X. We focus on the *global* configuration by exploiting the latent centers. We denote by Φ the function that maps a latent code z_i to its corresponding latent center. The center of the i^{th} sample is determined by the index $\arg\max_{j'}(p_{ij'})$. Samples from Θ are transformed into a decoded version of their associated centers. The output of $\mathcal{T}_1$ is described by:

$$x_i^{\text{clus}} = \begin{cases} \exp\Big(W_\mu \, f_D\big(\Phi(z_i)\big)\Big) & \text{if } i \in \Theta, \\ x_i & \text{otherwise.} \end{cases} \tag{8}$$

The second operation $\mathcal{T}_2$ gradually constructs A^{clus} from A. We focus on the *local* configuration by exploiting the latent topology captured by $\mathcal{G}^{\text{latent}}$. The initial structure A is transformed by adding and dropping edges. We add an edge between two nodes v_i and v_j if v_i and v_j belong to the set Θ and are connected in $\mathcal{G}^{\text{latent}}$. We drop an edge between two nodes v_i and v_j if v_i and v_j belong to the set Θ and have different clustering indices $\arg\max_{j'}(p_{ij'}) \neq \arg\max_{j'}(p_{jj'})$.

The two operations $\mathcal{T}_1$ and $\mathcal{T}_2$ play complementary roles. They explore global and local latent configurations to make progressive clustering-oriented corrections to the graph $\mathcal{G}$. The complete transformation $\mathcal{T}$ is provided in Algorithm 2.

The self-supervision task of the second phase consists of generating $\mathcal{G}^{\text{clus}}$. Accordingly, we minimize the negative log-likelihood of $p(\mathcal{G}^{gen} = \mathcal{G}^{\text{clus}} \mid Z)$. The resulting self-supervised function L_{SS} can be expressed as follows:

$$\begin{aligned} L_{\text{SS}}(\mathcal{G}^{\text{clus}}) &= -\log(p(\mathcal{G}^{gen} = \mathcal{G}^{\text{clus}} \mid Z)) \\ &= L_{X^{\text{clus}}} + L_{A^{\text{clus}}}, \end{aligned} \tag{9}$$

where $L_{X^{\text{clus}}} = -\log(p(\hat{X} = X^{\text{clus}} \mid Z))$ is the loss for generating the features X^{clus}, and $L_{A^{\text{clus}}} = -\log(p(\hat{A} = A^{\text{clus}} \mid Z))$ is the loss for generating the structure A^{clus}. The total loss function for the clustering phase is $L_{\text{SS}}(\mathcal{G}^{\text{clus}}) + \gamma \, L_{\text{PS}}$, where γ is a balancing hyperparameter.

Training Algorithm of the Proposed Method

We pretrain our model for R_1 iterations. The pretraining phase updates the parameters $\left\{W_E^{(l)}, W_D^{(l)}, W_\pi, W_\mu, W_\sigma\right\}$ to minimize the loss $L_{\text{SS}}(\mathcal{G})$ using the Adam optimizer. After that, we fine-tune the model until the number of reliable samples $|\Omega|$ covers 80% of the training set size. The clustering phase updates the parameters $\left\{W_E^{(l)}, W_D^{(l)}, W_\pi, W_\mu, W_\sigma, \Omega_j\right\}$ to minimize the loss $L_{\text{SS}}(\mathcal{G}^{\text{clus}}) + \gamma \, L_{\text{PS}}$ using the Adam optimizer. The training algorithm of our model and its computational complexity are provided in Appendix C and Appendix D, respectively.

Experiments

We perform several experiments on eight representative real-world scRNA-seq datasets collected by popular sequencing platforms from diverse species. The data description and preprocessing are discussed in Appendix E. We compare our method entitled scTPF (single-cell Topological and Probabilistic Filtering-based deep clustering) with state-of-the-art scRNA-seq data clustering approaches including two deep embedding methods: scGMAI (Yu et al. 2021) and scDCA (Eraslan et al. 2019); three deep clustering methods: scDCC (Tian et al. 2021), scDC (Tian et al. 2019), and scziDesk (Chen et al. 2020)); and four deep graph clustering methods: scGAE (Luo et al. 2021), scGNN (Wang et al. 2021), scTAG (Yu et al. 2022), and scDSC (Gan et al. 2022). All baselines are discussed in the Related Work section. We use the official code for each baseline and tune the hyperparameters if the authors do not provide specific recommendations. We evaluate the clustering performance based on three standard metrics, namely ACC, NMI, and ARI. We report the clustering results in %. Higher values imply better clustering quality. We also report the execution time of the evaluated methods in seconds. As described in Appendix G, all experiments are conducted with the same hardware and software setups. We estimate the level of FR and FD using the metrics Λ_{FR} and Λ_{FD} (Mrabah, Bouguessa, and Ksantini 2022), respectively. Higher values imply a lower level of FR (FD, respectively). We provide a full description of the evaluation metrics in Appendix H. In Appendix F, we specify and discuss the architecture, learning rates, and all other hyperparameters of our approach. Visualizations are provided in Appendix I.

Clustering Performance: In Table 1, we compare the clustering performance of our approach with several state-of-the-art methods. We run each model ten times and report the average values. As can be seen, the deep clustering methods generally achieve better results than the deep embedding ones. In addition, the deep graph clustering approaches generally yield better results than the deep clustering ones. These results confirm: (1) the relevance of performing joint embedding learning and clustering, and (2) the relevance of exploiting structural information from the gene expression count matrix during the encoding process. Most importantly, our approach outperforms all the other methods in terms of ACC, NMI, and ARI on all datasets. For example, the average difference in clustering performance between scTPF and the most competitive approach (i.e., scTAG) is greater than

Metric	Dataset	Deep Graph Clustering					Deep Clustering			Deep Embedding	
		scTPF	scTAG	scGNN	scGAE	scDSC	scziDesk	scDC	scDCC	scDCA	scGMAI
ACC	Muraro	**95.76**	89.35	66.30	89.75	86.52	73.66	75.12	84.35	73.61	61.40
	Plasschaert	**97.46**	89.01	53.56	83.83	91.76	93.56	52.60	61.34	62.66	48.07
	QX_LM	**99.71**	97.77	63.54	75.23	78.49	96.67	75.06	82.04	74.52	60.88
	QS_Diaph	**99.19**	98.39	68.04	55.51	94.94	97.82	71.84	71.26	72.30	47.93
	QS_Heart	**99.11**	96.13	63.50	89.46	92.05	94.46	55.53	68.23	70.72	60.09
	QS_LM	**99.63**	99.17	85.41	62.56	86.06	97.80	60.64	74.68	58.53	57.98
	Wang_Lung	**99.34**	95.14	97.97	77.11	96.02	97.59	98.97	90.51	98.62	38.56
	Young	**84.48**	81.35	50.30	70.72	67.88	79.81	58.75	67.41	64.57	42.00
	Average	**96.83**	93.28	68.57	75.52	86.71	91.42	68.56	74.97	71.94	52.11
NMI	Muraro	**88.53**	82.92	64.10	83.53	85.00	77.76	75.49	83.84	76.21	71.68
	Plasschaert	**90.10**	73.79	63.17	68.44	80.02	79.80	61.22	64.97	65.52	57.11
	QX_LM	**98.82**	93.64	74.43	79.39	80.61	91.28	84.05	87.98	83.69	76.39
	QS_Diaph	**96.49**	93.28	71.34	68.04	91.67	91.40	78.07	81.04	78.38	68.36
	QS_Heart	**96.57**	89.43	73.20	78.39	89.58	87.75	65.31	72.75	70.61	69.41
	QS_LM	**98.36**	96.44	80.35	72.71	82.73	92.15	70.48	79.94	71.71	71.98
	Wang_Lung	**92.77**	71.50	84.61	58.62	74.54	81.46	90.11	57.99	87.38	34.32
	Young	**81.83**	79.16	40.03	65.27	72.18	76.72	61.78	67.66	63.66	49.54
	Average	**92.93**	85.02	68.90	71.79	82.04	84.79	73.31	74.52	74.64	62.34
ARI	Muraro	**92.70**	87.02	49.19	88.21	89.51	66.30	66.09	74.10	64.59	51.32
	Plasschaert	**94.24**	76.97	46.40	70.64	86.99	85.88	40.70	49.76	50.69	57.11
	QX_LM	**99.58**	95.45	58.33	63.31	74.79	91.28	75.00	80.23	74.43	50.61
	QS_Diaph	**98.27**	96.37	56.09	42.90	96.15	93.69	64.79	66.03	64.98	41.11
	QS_Heart	**98.18**	94.09	56.74	84.89	94.78	92.08	46.73	58.18	59.87	43.68
	QS_LM	**99.25**	98.11	79.68	53.79	90.89	94.29	53.84	65.77	54.44	48.99
	Wang_Lung	**97.12**	80.14	91.31	59.98	83.48	89.74	95.50	64.10	94.02	13.25
	Young	**73.18**	70.61	28.98	58.87	54.61	66.81	44.69	52.85	50.06	32.87
	Average	**94.06**	87.34	58.34	65.32	83.90	85.00	60.91	63.87	64.13	42.36
Time	Muraro	103	**54**	733	61	88	106	74	83	240	152
	Plasschaert	413	**159**	1725	490	242	378	165	216	546	589
	QX_LM	116	**70**	816	189	120	278	73	82	198	321
	QS_Diaph	70	50	341	**46**	64	98	58	73	52	78
	QS_Heart	151	**87**	1475	221	134	406	164	273	261	300
	QS_LM	87	63	413	**46**	52	115	56	78	66	86
	Wang_Lung	739	519	1867	718	**150**	632	307	190	202	572
	Young	316	213	1243	351	**202**	601	223	203	432	367

Table 1: Clustering results on eight scRNA-seq datasets. Best methods in bold and second best underlined.

Method		Muraro				Plasschaert				Young			
FR mech.	**FD mech.**	**ACC**	**NMI**	**ARI**	**Time**	**ACC**	**NMI**	**ARI**	**Time**	**ACC**	**NMI**	**ARI**	**Time**
✗	✗	91.94	85.49	89.97	60	93.88	83.79	88.27	260	81.49	79.21	70.82	188
✗	✓	93.78	87.24	91.67	96	97.09	89.60	93.31	356	82.04	81.12	72.05	271
✓	✗	94.44	87.30	91.91	72	97.30	89.88	93.94	282	83.57	81.30	72.21	223
✓	✓	95.76	88.53	92.70	103	97.46	90.10	94.24	413	84.48	81.83	73.18	316

Table 2: Impact of the proposed FR and FD mechanisms on the clustering performance of scTPF.

6.7% in terms of NMI and ARI on all dataset. Unlike the other methods, our approach stands out by two key aspects: (1) the capacity to mitigate FR and FD, and (2) exploiting the interaction between local and global latent configurations to progressively control the clustering task. Despite the considerable improvement in clustering results, the execution time of our method remains competitive with several recent strategies such as scGMAI, scziDesk, scDCA, and scGNN.

FR and FD: In Figure 2, we illustrate the ability of our method to mitigate FR and FD. To this end, we introduce a variant of our model scTPF(-FR-FD), which comes without the FR and FD mechanisms. Removing the FR mechanism is achieved by computing (q_{ij}) using hard-clustering assignments directly instead of the gradual filtering method $\mathcal{F}$. Removing the FD mechanism is achieved by reconstructing the input graph $\mathcal{G}$ instead of the graph $\mathcal{G}^{\textbf{clus}}$. We compare scTPF

Method	Muraro			Plasschaert			QX_LM			Young		
FR mechanism	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**	**ACC**	**NMI**	**ARI**
scTPF ($\mathcal{F}_1$)	93.59	86.89	91.57	96.99	89.29	93.75	99.60	98.43	99.36	79.19	75.76	67.15
scTPF ($\mathcal{F}_2$)	94.06	87.04	91.80	97.12	89.65	93.31	99.57	98.39	99.42	82.06	81.10	71.99
scTPF ($\mathcal{T}_1$)	94.53	87.35	91.99	97.28	89.83	93.91	99.64	98.61	99.52	83.92	81.09	72.35
scTPF ($\mathcal{T}_2$)	94.39	87.34	91.95	97.23	89.70	93.78	99.61	98.47	99.47	83.43	80.87	72.17
scTPF	95.76	88.53	92.70	97.46	90.10	94.24	99.71	98.82	99.58	84.48	81.83	73.18

Table 3: Impact of $\mathcal{F}_1$, $\mathcal{F}_2$, $\mathcal{T}_1$, and $\mathcal{T}_2$ on the clustering performance of scTPF.

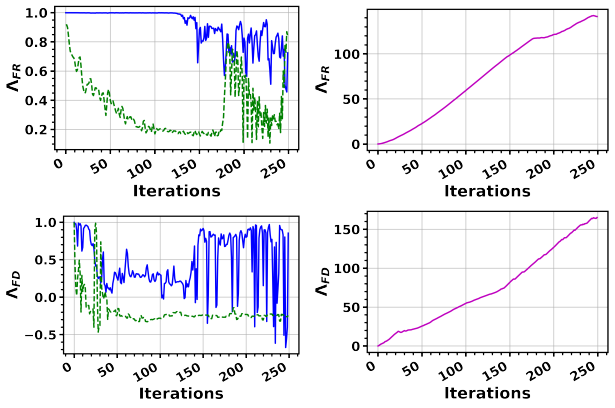

Figure 2: Performance of scTPF (in continuous blue line) and scTPF(-FR-FD) in (in dashed green line) in terms of Λ_{FR} and Λ_{FD} on the Young dataset. The first row of the second column computes the cumulative difference between Λ_{FR}(scTPF) and Λ_{FR}(scTPF(-FR-FD)). The second row of the second column computes the cumulative difference between Λ_{FD}(scTPF) and Λ_{FD}(scTPF(-FR-FD)).

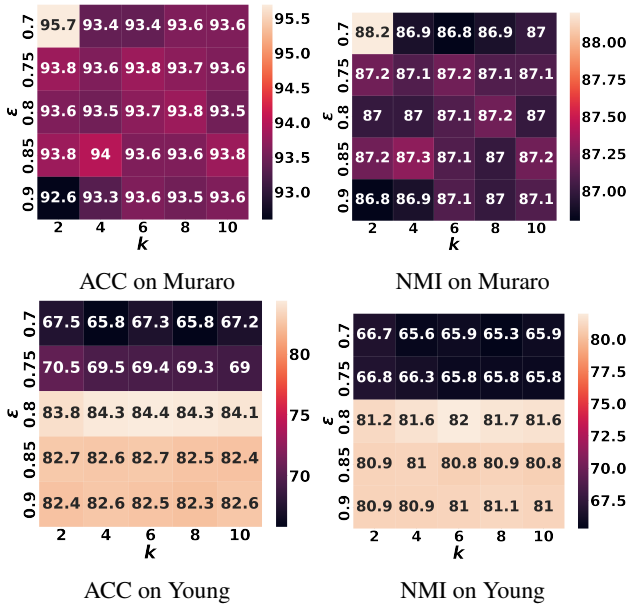

Figure 3: Sensitivity of scTPF to the hyperparameters ϵ and k in terms of ACC and NMI.

with its variant scTPF(-FR-FD) in terms of Λ_{FR} (Λ_{FD}, respectively) and the cumulative difference between the two models in terms of Λ_{FR} (Λ_{FD}, respectively). As we can see, the cumulative difference in terms of Λ_{FR} (Λ_{FD}, respectively) has a prominent increasing tendency during the training process. These results show that the proposed mechanisms $\mathcal{F}$ and $\mathcal{T}$ alleviate FR and FD, respectively.

Ablation Study: In Table 2, we illustrate the impact of our FR and FD mechanisms on clustering performance. First, we can see that scTPF produces better results than the other variants in terms of ACC, NMI, and ARI. These results confirm that the mitigation of FR and FD improves the clustering performance. Second, we find that adding a correction mechanism $\mathcal{T}$ leads to a longer computation time compared to the protection mechanism $\mathcal{F}$. In Table 3, we illustrate the impact of $\mathcal{F}_1$ and $\mathcal{F}_2$ ($\mathcal{T}_1$ and $\mathcal{T}_2$, respectively) on clustering performance. As we can see, exploiting the local and global configurations simultaneously to adjust the pseudo-supervised (self-supervised, respectively) task leads to better clustering results in terms of ACC, NMI, and ARI. These results confirm that exploring the interaction between local and global configurations improves clustering performance.

Sensitivity Analysis: In Figure 3, we illustrate the sensitivity of our model to the data-dependent hyperparameters (ϵ and k). We fix ϵ and k from the ranges $[0.7, 0.75, 0.8, 0.85, 0.9]$ and $[2, 4, 6, 8, 10]$, respectively, using grid search. The other hyperparameters and design choices are maintained at fixed values independently of the input dataset. As we can see, scTPF shows consistent results in terms of ACC and NMI for a wide range of values of ϵ and k.

Conclusion

This paper proposes a graph embedding method that can effectively identify cell clusters from single-cell RNA sequencing data. Our method explores the interaction between local and global latent configurations to adjust the pseudo-supervised and self-supervised tasks. In particular, we propose a topological and probabilistic filter $\mathcal{F}$ that mitigates the effect of Feature Randomness. Furthermore, we introduce a cell-cell graph structure and content correction mechanism $\mathcal{T}$ that counteracts the effect of Feature Drift. Through relevant empirical results, we show that our approach outperforms multiple state-of-the-art methods by alleviating the effect of Feature Randomness and Feature Drift and exploiting the interaction between local and global latent configurations.

References

Bhaduri, A.; Di Lullo, E.; Jung, D.; Müller, S.; Crouch, E. E.; Espinosa, C. S.; Ozawa, T.; Alvarado, B.; Spatazza, J.; Cadwell, C. R.; et al. 2020. Outer radial glia-like cancer stem cells contribute to heterogeneity of glioblastoma. *Cell stem cell*, 26(1): 48–63.

Chen, L.; Wang, W.; Zhai, Y.; and Deng, M. 2020. Deep soft K-means clustering with self-training for single-cell RNA sequence data. *NAR genomics and bioinformatics*, 2(2): lqaa039.

Eraslan, G.; Simon, L. M.; Mircea, M.; Mueller, N. S.; and Theis, F. J. 2019. Single-cell RNA-seq denoising using a deep count autoencoder. *Nature communications*, 10(1): 1–14.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, 226–231.

Gan, Y.; Huang, X.; Zou, G.; Zhou, S.; and Guan, J. 2022. Deep structural clustering for single-cell RNA-seq data jointly through autoencoder and graph neural network. *Briefings in Bioinformatics*, 23(2): bbac018.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural Message Passing for Quantum Chemistry. In *NeurIPS*, volume 70, 1263–1272.

Huang, M.; Wang, J.; Torre, E.; Dueck, H.; Shaffer, S.; Bonasio, R.; Murray, J. I.; Raj, A.; Li, M.; and Zhang, N. R. 2018. SAVER: gene expression recovery for single-cell RNA sequencing. *Nature methods*, 15(7): 539–542.

Jindal, A.; Gupta, P.; Sengupta, D.; et al. 2018. Discovery of rare cells from voluminous single cell expression data. *Nature communications*, 9(1): 1–9.

Johnson, S. C. 1967. Hierarchical clustering schemes. *Psychometrika*, 32(3): 241–254.

Kipf, T. N.; and Welling, M. 2016. Variational graph auto-encoders. In *NeurIPS workshop*, volume 32, 1–3.

Kipf, T. N.; and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Kiselev, V. Y.; Andrews, T. S.; and Hemberg, M. 2019. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nature Reviews Genetics*, 20(5): 273–282.

Li, W. V.; and Li, J. J. 2018. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications*, 9(1): 1–9.

Lin, P.; Troup, M.; and Ho, J. W. 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome biology*, 18(1): 1–11.

Liu, Y.; Tu, W.; Zhou, S.; Liu, X.; Song, L.; Yang, X.; and Zhu, E. 2022. Deep Graph Clustering via Dual Correlation Reduction. In *AAAI*, volume 36, 7603–7611.

Luo, Z.; Xu, C.; Zhang, Z.; and Jin, W. 2021. A topology-preserving dimensionality reduction method for single-cell RNA-seq data using graph autoencoder. *Scientific reports*, 11(1): 1–8.

Maennel, H.; Alabdulmohsin, I. M.; Tolstikhin, I. O.; Baldock, R.; Bousquet, O.; Gelly, S.; and Keysers, D. 2020. What Do Neural Networks Learn When Trained With Random Labels? *NeurIPS*.

McInnes, L.; Healy, J.; and Melville, J. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Mrabah, N.; Bouguessa, M.; and Ksantini, R. 2022. Adversarial Deep Embedded Clustering: On a Better Trade-off Between Feature Randomness and Feature Drift. *TKDE*, 34(04): 1603–1617.

Paik, D. T.; Cho, S.; Tian, L.; Chang, H. Y.; and Wu, J. C. 2020. Single-cell RNA sequencing in cardiovascular development, disease and medicine. *Nature Reviews Cardiology*, 17(8): 457–473.

Papalexi, E.; and Satija, R. 2018. Single-cell RNA sequencing to explore immune cell heterogeneity. *Nature Reviews Immunology*, 18(1): 35–45.

Park, S.; and Zhao, H. 2018. Spectral clustering based on learning similarity matrix. *Bioinformatics*, 34(12): 2069–2076.

Saelens, W.; Cannoodt, R.; Todorov, H.; and Saeys, Y. 2019. A comparison of single-cell trajectory inference methods. *Nature biotechnology*, 37(5): 547–554.

Stubbington, M. J.; Rozenblatt-Rosen, O.; Regev, A.; and Teichmann, S. A. 2017. Single-cell transcriptomics to explore the immune system in health and disease. *Science*, 358(6359): 58–63.

Tian, T.; Wan, J.; Song, Q.; and Wei, Z. 2019. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nature Machine Intelligence*, 1(4): 191–198.

Tian, T.; Zhang, J.; Lin, X.; Wei, Z.; and Hakonarson, H. 2021. Model-based deep embedding for constrained clustering analysis of single cell RNA-seq data. *Nature communications*, 12(1): 1–12.

van Dijk, D.; Nainys, J.; Sharma, R.; Kaithail, P.; Carr, A. J.; Moon, K. R.; Mazutis, L.; Wolf, G.; Krishnaswamy, S.; and Pe'er, D. 2017. MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data. *BioRxiv*, 111591.

Wang, B.; Zhu, J.; Pierson, E.; Ramazzotti, D.; and Batzoglou, S. 2017. Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature methods*, 14(4): 414–416.

Wang, J.; Ma, A.; Chang, Y.; Gong, J.; Jiang, Y.; Qi, R.; Wang, C.; Fu, H.; Ma, Q.; and Xu, D. 2021. scGNN is a novel graph neural network framework for single-cell RNA-Seq analyses. *Nature communications*, 12(1): 1–11.

Welch, J. D.; Kozareva, V.; Ferreira, A.; Vanderburg, C.; Martin, C.; and Macosko, E. Z. 2019. Single-cell multiomic integration compares and contrasts features of brain cell identity. *Cell*, 177(7): 1873–1887.

Yu, B.; Chen, C.; Qi, R.; Zheng, R.; Skillman-Lawrence, P. J.; Wang, X.; Ma, A.; and Gu, H. 2021. scGMAI: a Gaussian mixture model for clustering single-cell RNA-Seq data based on deep autoencoder. *Briefings in Bioinformatics*, 22(4): 316.

Yu, Z.; Lu, Y.; Wang, Y.; Tang, F.; Wong, K.-C.; and Li, X. 2022. ZINB-Based Graph Embedding Autoencoder for Single-Cell RNA-Seq Interpretations. *AAAI*, 36(4): 4671–4679.