Published online 9 October 2020

NAR Genomics and Bioinformatics, 2020, Vol. 2, No. 4 **1**
doi: 10.1093/nargab/lqaa082

scAIDE: clustering of large-scale single-cell RNA-seq data reveals putative and rare cell types

Kaikun Xie[1,2,†], **Yu Huang**[1,2,†], **Feng Zeng**[3,4], **Zehua Liu**[5,6,7] **and Ting Chen**[1,2,*]

[1]Institute for Artificial Intelligence, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, [2]Tsinghua-Fuzhou Institute of Digital Technology, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China, [3]Department of Automation, Xiamen University, Xiamen 361005, China, [4]National Institute for Data Science in Health and Medicine, Xiamen University, Xiamen 361005, China, [5]Center for Computational and Integrative Biology, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, USA, [6]Department of Molecular Biology, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, USA and [7]Broad Institute of Massachusetts Institute of Technology and Harvard, Cambridge, MA 02142, USA

Received May 09, 2020; Revised August 20, 2020; Editorial Decision September 10, 2020; Accepted September 18, 2020

ABSTRACT

Recent advancements in both single-cell RNA-sequencing technology and computational resources facilitate the study of cell types on global populations. Up to millions of cells can now be sequenced in one experiment; thus, accurate and efficient computational methods are needed to provide clustering and post-analysis of assigning putative and rare cell types. Here, we present a novel unsupervised deep learning clustering framework that is robust and highly scalable. To overcome the high level of noise, scAIDE first incorporates an autoencoder-imputation network with a distance-preserved embedding network (AIDE) to learn a good representation of data, and then applies a random projection hashing based *k*-means algorithm to accommodate the detection of rare cell types. We analyzed a 1.3 million neural cell dataset within 30 min, obtaining 64 clusters which were mapped to 19 putative cell types. In particular, we further identified three different neural stem cell developmental trajectories in these clusters. We also classified two subpopulations of malignant cells in a small glioblastoma dataset using scAIDE. We anticipate that scAIDE would provide a more in-depth understanding of cell development and diseases.

INTRODUCTION

The advancement in single-cell RNA-sequencing technology has grown exponentially in terms of sample sizes and accuracy (1–3). Identifying different cell types and subtypes remains one of the initial core analysis in single-cell data, prior to further downstream analysis. As the amount of data increases, we can gain a more holistic view of the identity and functionality of each cell. With recent large-scale pilot studies such as the Human Cell Atlas (4), unsupervised scalable and accurate computational approaches are essential for identifying different cell types.

Although many different computational methods have been developed to cluster and classify single-cell datasets, there is a trade-off between computational time and accuracy. Briefly, classification approaches are fast in analysis (provided that the model has been trained) with relatively reliable accuracy. However, such methods (5–7) require prior knowledge and labeled datasets for training (8). On the other hand, most clustering methods follow the pipeline of (i) clustering into groups, (ii) identify significantly expressed genes and (iii) manually validate marker genes with cell types. In general, there are three main categories of clustering methods. Traditional methods, including SC3, pcaReduce, Seurat, SIMLR and various others, rely on conventional dimensionality reduction methods such as PCA or t-SNE and then apply k-means or graph-based clustering to identify clusters (9–13). Secondly, iterative methods such as BackSPIN, SAIC and Panoview (14–16), attempt to provide a hierarchical structure over the identified cell clusters. More recent studies focus on developing deep learning methods to model the dropout events and provide nonlinear dimensionality reduction to represent single-cell datasets better (17–20). Despite the efforts in developing efficient and complex models, the clustering performance can be severely affected by hyper-parameters and varies accordingly in different datasets (21).

*To whom correspondence should be addressed. Tel: +86 135 1100 5758; Email: tingchen@tsinghua.edu.cn
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
Present address: Department of Computer Science and Technology, Tsinghua University, Beijing, China.

Identifying rare cell types is important in dissecting the cellular heterogeneity in global cell population. Recent methods, including RaceID, CellSIUS and GiniClust3, utilize clustering steps followed by an assignment step to identify rare cell types (22–25). In particular, CellSIUS first partitions the cells into coarse clusters and then identifies rare cell subpopulations based on correlated genes sets with respect to each subpopulation. The benefit of CellSIUS is that upregulated genes sets could be obtained for the identified cell types. Another recent method, FiRE (26), developed an algorithmic approach by directly assigning a rareness score to each cell without clustering.

In this manuscript, we propose a fully unsupervised deep learning clustering analysis framework, scAIDE (Figure 1A). First, we implemented an autoencoder-imputed distance-preserved embedding network (AIDE) to obtain a good representation of single-cell data which separates different cell types well. Subsequently, to identify small or rare cell types as well as common cell types, we developed a random projection hashing based k-means algorithm (RPH-kmeans). We can also automatically detect the number of clusters based on RPH-kmeans. Moreover, we provide a systematic biological analysis on the annotation of cell types. The performance and stability of scAIDE are extensively compared to existing state-of-the-art methods on seven single-cell datasets across different sequencing protocols. We further applied our clustering framework to analyze cell subpopulations in a small tumor dataset, a 68k peripheral blood mononuclear cells (PBMC), and finally, a 1.3 million neural dataset (27). We were able to identify small distinct cell populations, such as Cajal-Retzius cells (accounting for about 1.6% of the total population, expressing *Reln* and *Tbr1*), which are important to modulate early cortical patterning (28). In general, scAIDE is a scalable and efficient clustering framework which is consistent when applied to different single-cell datasets.

MATERIALS AND METHODS

Overview of scAIDE

There are two main components in scAIDE, namely AIDE for dimensionality reduction and RPH-kmeans for clustering, as shown in Figure 1A. Subsequently, we developed a general pipeline to provide biological analysis using marker genes and visualization of possible cell type development based on AIDE embedding.

Most clustering methods include a variable gene selection process to reduce the matrix to a reasonable size. However, to retain most information, we believe that the full gene expression should be used with minimal pre-processing (Supplementary Note VI and Table S25). We first filtered cells and genes with a minimum count of 1, followed by cell normalization and log transformation used in Scanpy and Seurat (11,29). As a result, most of the datasets contained about 10 000–20 000 genes after pre-processing (Table 1).

Autoencoder-imputed distance-preserved embedding (AIDE)

The architecture of AIDE consists of an imputation module and a dimension reduction module, as shown in Figure 1A. In the imputation module, the gene expression vector is fed to an autoencoder (AE) to correct biological noise such as dropout events. As AE captures the important latent structure of the data in the hidden layer and learns to regenerate the data, it is a natural extension that we can recover an imputed expression vector. We also added dropout layers in AE to avoid overfitting. Considering that the hidden vectors produced by simple AE may not be suitable for Euclidean-based clustering methods (e.g., k-means), we developed a fully connected network called multidimensional scaling (MDS) encoder in the dimension reduction module. The MDS encoder represents the dissimilarity/distance between the imputed expression vectors (produced by AE) as the Euclidean distance between projected points in a low-dimensional space. This matches with our latter developed random projection hashing based clustering algorithm which is also Euclidean-based.

Specifically, let $D = \{\boldsymbol{x}_i\}_{i=1}^{N}$ be the dataset with $\boldsymbol{x}_i \in R^G$ denoting the gene expression vector of cell i, G denotes the number of genes and $D_p = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) | \boldsymbol{x}_i, \boldsymbol{x}_j \in D\}$ denotes all the cell pairs. The reconstruction loss of AE is defined as:

$$L_{\text{rec}} = \frac{1}{2\left|D_p\right|} \sum_{(\mathrm{x}_i, \mathrm{x}_j) \in D_p} \left(\left|\left| \mathbf{x}_i - \hat{\mathbf{x}}_i \right|\right|_2^2 + \left|\left| \mathbf{x}_j - \hat{\mathbf{x}}_j \right|\right|_2^2 \right),$$

where $\hat{\boldsymbol{x}} = f_a(\boldsymbol{x}; \boldsymbol{W}_a)$ is also a G-dimensional vector with imputed gene expression, and $\boldsymbol{W}_a$ denotes the learnable weights of the AE, f_a. MDS (30) is a dimension reduction technique that preserves the dissimilarity/distance between pairs of objects. Here, we developed a neural network adaption of MDS to generate a low-dimensional representation of $\boldsymbol{x}$ by preserving the dissimilarity between $\hat{\boldsymbol{x}}$. The corresponding loss is defined as follows:

$$L_{\text{mds}} = \frac{1}{\left|D_p\right|} \sum_{(\boldsymbol{x}_i, \boldsymbol{x}_j) \in D_p} \left| ||\boldsymbol{z}_i - \boldsymbol{z}_j||_2^2 - \varphi\left(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j\right)^2 \right|$$

where $\boldsymbol{z} = f_m(\boldsymbol{x}; \boldsymbol{W}_m)$ is a d-dimensional vector ($d \ll G$), and $\boldsymbol{W}_m$ is the learnable weight of MDS encoder, f_m. φ denotes a specific dissimilarity/distance metric in the space of the imputed gene expression, and we used the Euclidean distance in this paper:$\varphi(\hat{\boldsymbol{x}}_i, \hat{\boldsymbol{x}}_j) = ||\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{x}}_j||_2$.

The training of AIDE can be divided into two stages: pre-training and joint tuning. In the pre-training stage, parameters of AE are optimized by minimizing $L_{\text{rec}}(D_p; \boldsymbol{W}_a)$. In the joint tuning phase, both the AE and MDS encoder are trained by minimizing

$$L\left(D_p; \boldsymbol{W}_a, \boldsymbol{W}_m\right) = L_{\text{rec}}\left(D_p; \boldsymbol{W}_a\right) + \alpha L_{\text{mds}}\left(D_p; \boldsymbol{W}_a, \boldsymbol{W}_m\right),$$

where $\alpha > 0$ is the coefficient that controls the relative weights of L_{rec} and L_{mds}, which also affects the degree of fitness of AE. Note that there is no need to generate all possible combinations of cell pairs (D_p) for AIDE training. In practice, we feed the model with a mini-batch of cell pairs randomly selected from dataset D iteratively until the training converges. After training, we use the MDS encoder to generate the embeddings $\{\boldsymbol{z}_i\}_{i=1}^{N}$, taking the gene expression vectors $\{\boldsymbol{x}_i\}_{i=1}^{N}$ as input.

We further compared the performance of AIDE against single components of AE and MDS encoder to show the

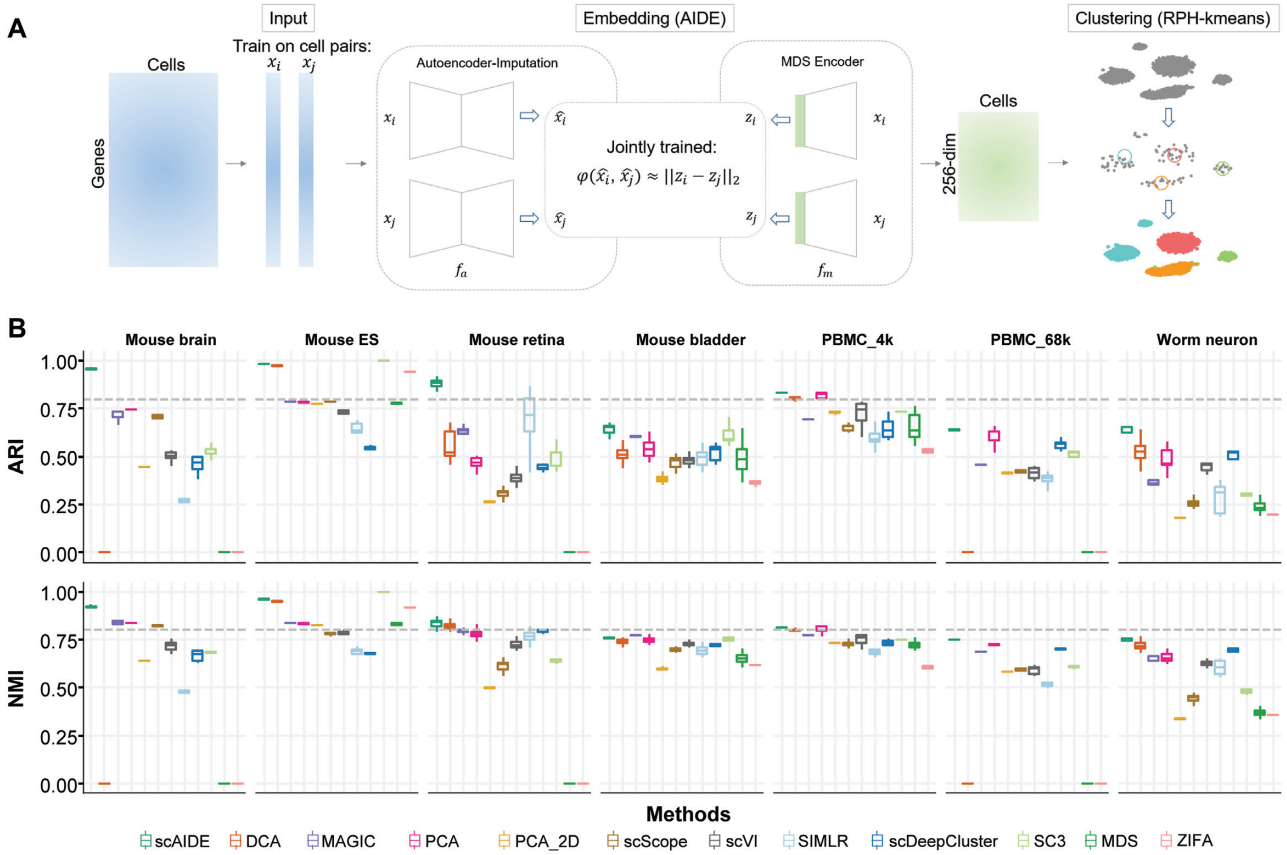

Figure 1. Overview and performance of scAIDE. (**A**) A schematic overview of the architecture of scAIDE. (**B**) The overall comparison of ARI and NMI performance on seven single-cell datasets. The dotted gray line represents a threshold of 0.8. For scDeepCluster and SC3, each algorithm was run five times to generate a distribution of results. For methods that involved a dimension reduction step, five embeddings were generated with the same parameters, and we obtained the boxplot by applying clustering to each embedding 10 times. We set k as the number of known cell type labels for comparison. Observations at 0 indicate that the experiment was not performed either because of insufficient memory or running time was >4 h.

Table 1. Description of analyzed datasets

Datasets	No. of cells	No. of genes used	No. of cell types	Group size (min, max)	Technology (reads/cell)
Brain 1.3m	1 300 774	23 909	-	-	10× V2 (18 500)
Mouse brain	160 796	20 803	7	(1826, 74 539)	10× V1
PBMC 68k	68 579	20 387	10	(176, 21 429)	10× V1 (20 000)
PBMC 4k	4271	16 653	8	(135, 1292)	10× V2 (87 000)
Mouse bladder	2746	19 771	16	(13, 717)	Microwell-seq
Mouse retina	27 499	13 166	19	(48, 10 888)	Drop-seq (8200)
Mouse ES	2717	24 047	4	(303, 933)	Droplet-based
Worm neuron	4186	13 488	10	(70, 1015)	sci-RNA-seq
Jurkat 293T cells	1580	1000	2	(40, 1540)	10×
MGH107 (WHO II)	252	23 686	3	(4, 98)	Smart-seq2
Simulation datasets	5000	10 000	10	Approx. (68, 1099)	

outperformance of the novel architecture (Supplementary Figure S1, Note II and Tables S12–13).

Random projection hashing-based *k*-means clustering (RPH-kmeans)

As k-means is simple with low time complexity, many studies adapt it to cluster single-cell RNA-seq data (9,10,22). However, one major downside is that k-means is highly sensitive to initial cluster centers. Thus, when the size of the underlying cluster groups is highly imbalanced, which is often the case with single-cell data (Table 1), the resulting clusters become biased toward larger cell populations. The standard random initialization and the most popular *k*-means++ (31) initialization strategy both choose initial centers located in large-size groups with extremely high probability. As a result, the cluster centers will be stuck in large groups since small groups have little impact on calculating new centers during each iteration. The larger groups may eventually be partitioned into several parts, leading to poor clustering performance. (Supplementary Figure S2).

In order to solve the data imbalance problem, we proposed a random projection hashing based k-means termed

RPH-kmeans, which initializes the cluster centers using one of the locality sensitive hashing (LSH) (32) techniques. The key principle of LSH is to project close data points to the same bucket with high probability. D. Datar *et al*. (33) proposed an LSH family for l_p distance metric. When p is 2 (the distance between two data points is evaluated by the Eulidean metric), the random projection-based hashing (RPH) function that maps a data point $\boldsymbol{v} \in R^d$ to an integer is defined as:

$$h_{\boldsymbol{a},b}\ (\boldsymbol{v}) = \left\lfloor \frac{\boldsymbol{a} \cdot \boldsymbol{v} + b}{w} \right\rfloor,$$

where $\boldsymbol{v} \in R^d$ denotes a data point, $\boldsymbol{a} \in R^d$ is a random vector with a_i drawn i.i.d. from the standard Gaussian distribution $N(0, 1)$, b is a random variable drawn from the uniform distribution $U(0, w)$, and w denotes the quantization step. Next, a composite hash function $g(\boldsymbol{v})$ is constructed by combining l hash functions:

$$g\ (\boldsymbol{v}) = (h_1(\boldsymbol{v}), \cdots, h_l(\boldsymbol{v}))$$

Thus, given a data point $\boldsymbol{v}$, the LSH function g will project $\boldsymbol{v}$ to an integer hash code vector. Data points are considered to be hashed into the same bucket if their hashed code vectors are exactly the same. In general, the closer (evaluated by the Euclidean distance) two data points are, the more likely they will be hashed into the same bucket.

The pipeline of cluster center initialization of RPH-kmeans can be summarized in two phases. In the first phase, the number of data points is reduced iteratively using LSH. In each iteration, the data points hashed to the same bucket will be merged to a weighted point. Finally, a data skeleton with much fewer points is generated. In the second phrase, weighted k-means (Algorithm S2) with k-means++ initialization will be applied to the skeleton to produce initial centers for RPH-kmeans. Since the number of points in large groups is significantly reduced, potential bias caused by data imbalance can be alleviated (as shown in Supplementary Figure S3). The pseudocode for RPH-kmeans initialization is described in Algorithm 2, and the full RPH-kmeans algorithm is described in Algorithm 1.

Due to the random property/character of LSH, error may be induced when generating the skeleton. For example, data points belonging to two large close groups are likely to be hashed into the same bucket, resulting in a poorly represented skeleton. Here, we provide two optional bucket correction strategy to solve the problem. Inspired by DACE (34), we first developed a radius-based strategy (Algorithm S3). Each bucket will be divided into sub-buckets by successively assigning every point to its closest center. If the distance to the center is greater than a given radius r, a new center will be created. The other one is called a size-based strategy (Algorithm S4). It retains only partial buckets with a small size because large buckets usually lead to the potential merging of two different groups.

Based on the weighted skeleton points generated by random projection, we further developed a weighted Bayesian Information Criterion (BIC) approach to estimate the number of clusters in the dataset (Supplementary Note I, Table S1, and Figure S4).

We noticed that the framework proposed by Li *et al*. (35) is similar to RPH-kmeans. However, they focused on using LSH to speed up k-means. To the best of our knowledge, we are the first to use LSH to approach the data imbalance problem in clustering.

Algorithm 1 RPH-kmeans

Input: Data $\{\mathbf{x}_i\}_{i=1}^N$, number of clusters k, repeat times r.
Output: Predicted labels $\mathbf{y}$, cluster centers $\mathbf{C}$, inertia ϕ.
  for $t=1$ **to** r **do**
    Initialize cluster centers $\{\mathbf{c}_i\}_{i=1}^k$ with algorithm 2;
    Run standard k-means and get inertia ϕ
  end for
  Return clustering result with smallest ϕ;

Algorithm 2 Centers initialization with RPH

Input: Data $\{\mathbf{x}_i\}_{i=1}^N$, number of clusters k, quantization step w, number of LSH functions l, max number of skeleton points m, max number of iteration T, bucket correction strategy f.
Output: Initial centers $\{\mathbf{c}_i\}_{i=1}^k$
  Skeleton $S=\{(\mathbf{x}_i,1)\}_{i=1}^N$, iteration $t=0$;
  while $t<T$ and $|S|>m$ **do**
    Randomly generate $g_{t,w}(\mathbf{v})=(h_{1,w}(\mathbf{v}),\cdots,h_{l,w}(\mathbf{v}))$;
    Use $g_{t,w}$ to partition S into buckets $\{B_i'\}_{i=1}^{u'}$;
    Generate corrected buckets: $\{B_i\}_{i=1}^{u}=f(\{B_i'\}_{i=1}^{u'})$;
    $S=\emptyset$
    for $i=1$ **to** u **do**
      $w=\sum_{(\mathbf{v}_j,w_j)\in B_i} w_j$;
      $\mathbf{v}=\frac{1}{w}\sum_{(\mathbf{v}_j,w_j)\in B_i} w_j\mathbf{v}_j$;
      $S=S\cup\{(\mathbf{v},w)\}$;
    end for
    $t=t+1$;
  end while
  Generate cluster centers $\{\mathbf{c}_i\}_{i=1}^k$ by clustering S using weighted k-means (algorithm S2) with k-means++ initialization.

Evaluation metrics

All clustering results are measured by the adjusted rand index (ARI) (36) and normalized mutual information (NMI) (37). Given two partitions U and V, let n_{ij} be the number of common data points of groups $u_i \in U$, and $v_j \in V$. ARI is defined as:

$$\mathrm{ARI} = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\Big/\binom{n}{2}}{\frac{1}{2}\left[\sum_i\binom{n_{i\cdot}}{2}+\sum_j\binom{n_{\cdot j}}{2}\right] - \sum_i\binom{n_{i\cdot}}{2}\sum_j\binom{n_{\cdot j}}{2}\Big/\binom{n}{2}}$$

where $n_{i\cdot} = \sum_j n_{ij}$, $n_{\cdot j} = \sum_i n_{ij}$.

NMI is defined as:

$$\mathrm{NMI} = \frac{\sum_{i,j}\frac{u_i\cap v_j}{n}\log\frac{n|u_i\cap v_j|}{|u_i||v_j|}}{\frac{1}{2}\left(-\sum_i\frac{|u_i|}{n}\log\frac{|u_i|}{n} - \sum_j\frac{|v_j|}{n}\log\frac{|v_j|}{n}\right)}$$

where n is the number of data points.

Data visualizations and biological analysis

In order to visualize the distribution of cluster groups and the embedding of scAIDE, we used t-stochastic neighboring embedding (t-SNE) for all our visualizations. The default parameters are applied without tuning using the R package, Rtsne.

For the discovery of marker genes, we first calculated the Wilcoxon's rank-sum test for each gene in the cluster. Then the log fold change values were measured to ensure that the identified marker gene is supported by sufficient samples. The threshold cut-off for the rank-sum test is set to a small value near 0 (for a strict detection of a small number of marker genes) and 1.5 for fold-change. Fold-change values were calculated as the ratio between group average gene expressions. We are only interested in the up-regulation of markers within a specific cluster, compared to the remaining cells.

In some current studies, cell types are assigned according to a few top marker genes. We believe that developing a systematic approach to assign cell types would be more reliable. To classify the cell types in the clustering analysis, we use gene markers from previous studies (38) and a single-cell gene marker database (39). We used a simple matching rate and the Jaccard index to quantify the number of overlapping marker genes. To test the significance of the assigned cell type, we implemented an enrichment *P*-value based on a hypergeometric distribution (40). After filtering, we define the total number of genes, G as the number of background genes. Suppose α denotes the number of identified markers from a particular cluster, and b the number of markers for a specific cell type, the number of overlapping genes is regarded as k. The enrichment *P*-value was calculated as follows:

$$p = \sum_{i=k}^{\min(a,\, b)} \frac{C_i^a \cdot C_{b-i}^{N-a}}{C_b^N}$$

Additionally, manual validations were also made by comparing specific top markers with existing studies.

Identifying possible trajectory development is one of the important downstream analysis in single-cell data. We utilize the AIDE embedding vectors to reflect the development of cell clusters. The intuition behind this is that cells from a similar lineage would be closer. First, we calculate the average AIDE vector for each respective cluster, and then we apply the Euclidean distance to clusters to obtain a k by k matrix, where k is the number of clusters. Then we perform a simple hierarchical clustering (with complete linkage) to reflect the relationship between cell clusters. Finally, we visualize the cell clusters using heatmap and dendrogram to depict the groupings of possible trajectory development.

Datasets

Real datasets. We used a total of 9 real single-cell datasets to quantify the performance of scAIDE in clustering analysis. The summary of each dataset is listed in Table 1, spanning across different sequencing technologies and varying dropout rates.

Simulation datasets. Concerning the rate of dropout events in more efficient sequencing technologies, it is necessary to develop robust clustering methods that can be generalized. We compared the current methods on multiple simulated datasets using splatter (41), ranging from 60% of dropouts to 93% dropouts (hereby referred to as sparsity). We simulated single-cell datasets with 5000 cells and 10 000 genes, with a highly imbalanced cell group assignment. The smallest cluster contained about 1.5%, while the largest contained roughly 23.2% of cells. Simulation parameters were obtained from a pre-processed smart-seq2 single-cell dataset of the development in mouse embryonic cells (42). The sparsity was tuned by altering the dropout parameters in splatter. The performance of the algorithms is evaluated using pre-determined group labels.

Additionally, we simulated another mouse embryonic dataset (43) by increasing the dropout sparsity of the dataset. We followed the pipeline of simulating dropout as used in splatter. The dataset had an original sparsity of about 70%; then, we simulated the data for 85, 90 and 96%.

RESULTS

Quantitative evaluation of scAIDE

To benchmark the general performance of scAIDE, we compared it to multiple state-of-the-art methods. This includes simple baselines such as MDS, PCA and PCA_2D followed by k-means, complex approaches like SC3 (9), SIMLR (12), MAGIC (44) and numerous deep learning methods: DCA (17), scDeepCluster (19), ZIFA (20), scVI (45) and scScope (46) (Figure 1B). One interesting observation was that PCA performed well when the gene expression was reduced to 256 components (PCA) instead of two components (PCA_2D) before applying the k-means clustering. Although PCA_2D (with two components) is commonly used as the baseline in many studies, we argue that significantly better results could be achieved by simply increasing the number of components (which better captures the information of the data). Thus, we hypothesized that a good representation of the gene expression data might lead to profound biological insights, even with simple clustering methods.

We used ARI and NMI to quantify how similar the clustering results are to the cell labels given in their respective original studies. A total of seven single-cell datasets were used to evaluate the performance, ranging from very distinct cell populations (for example, mouse embryonic stem cells) to very diverse populations (such as neural cells and PBMCs). Although some labels were not of the gold standard, these datasets provide a general baseline to compare the performance of current state-of-the-art clustering methods. scAIDE first reduces the gene expression input to a reduced AIDE embedding of 256-dimensions, followed by our developed RPH-kmeans clustering approach. In order to maintain a fair comparison for deep learning methods, we followed their pre-processing steps for each respective method. Also, we tested both their default embedding dimensions, and parameters which had similar model complexity to AIDE (Supplementary Note III and Tables S14–21). We plotted the best results and compared their performance in Figure 1B and Supplementary Note III. Over-

all, scAIDE demonstrates high stability and overall performance over current methods. In particular, our method outperformed other methods significantly in the mouse retina dataset (47), which contained 19 cell types with a total of 27 499 cells. Although it is highly imbalanced (the smallest group only contained 48 cells), our approach achieved a high average of 0.875 ARI and 0.825 NMI. Additionally, we also show that RPH-kmeans improves the clustering performance by benchmarking against *k*-means++ and *k*-means (with random initialization) on these seven datasets (Supplementary Note IV and Table S22).

We kept a small number of parameters for convenience and provided default parameters which demonstrate consistent performance across datasets (Supplementary Figure S5 and Table S2). We demonstrate that scAIDE is relatively stable using default parameters with little tuning required; the details for the parameters used in the experiments are included in the supplementary materials (Supplementary Tables S2-S6 and Note III).

Robustness of scAIDE under different dropout simulations

Although more accurate sequencing technology is emerging, cost and efficiency still constrain deep sequencing. Scalable technology, such as 10×, sequences at an average of 10–20k reads per cell, while smart-seq2 can sequence up to millions of reads per cell but at a higher cost. Thus, it is vital that computational methods can accurately recover the cell-type populations at various dropout rates.

We first evaluated the performance on fully simulated datasets using splatter (41), ranging from roughly 60% of zero expressions (similar proportion as seen in smart-seq2 datasets) to 93% (similar to 10× datasets). We simulated 5000 single-cell profiles of 10 000 genes, with 10 imbalanced cell groups (Figure 2A). In cases of lower sparsity levels (60–85%), the performance of scAIDE was on par with the consensus and imputation methods, SC3 and MAGIC; while outperforming other deep learning methods. scDeepCluster performed well in the two simulation cases with higher dropout rates (90 and 93%), despite its moderate performance on real datasets. This could be due to the use of a zero-inflated negative binomial (ZINB) layer in its model.

In order to further validate this result, we generated more realistic simulations by adding dropout events based on a logistic regression model (41). We used a well-defined reference dataset that consists of mouse embryonic stem cells (Mouse ES (43)) with four leukemia inhibitory factor (LIF) withdrawal time interval labels (day 0, 2, 4 and 7). SC3 was used as a baseline for comparison due to its superior performance on this dataset. We also included scDeepCluster as it performed relatively well in the previous simulation experiments. In Figure 2B, each algorithm was run five times, and their respective performance indexes were plotted. Despite the good performance of scDeepCluster in the previous simulated datasets, it fails to separate the cells here. scAIDE is highly consistent in all scenarios and outperforms SC3 when sparsity level exceeds 90%. Furthermore, the AIDE embedding clearly separates cells from the four different intervals, even in cases of high dropout events (Figure 2C). We show that scAIDE has a high potential to separate and delineate cell groups regardless of dropout situations.

Identification of putative and rare subpopulations in single-cell datasets

De novo clustering analysis has the potential to provide biological insight into the identification of rare cell types. In general, there are two important aspects in accurately separating different cell types and identifying rare subpopulations. First, cells should be well-represented in low dimensions; subsequently, clustering algorithms should accommodate the identification of small groups of cells. In simulation experiments, we first depict that the AIDE embedding is capable of separating different cell types, and that RPH-kmeans is tailored for the detection of rare cell types. Then, we applied scAIDE to three different datasets (case studies) to reveal novel biological findings. Particularly, not only did we identify different subpopulations within each dataset, but we also identified primed differentiation development of cell types.

Rare cell type detection in simulated datasets. Following Zheng's study (3) and a rare cell type detection method named FiRE (26), we mixed 2.5% (40 cells) of Jurkat cells into an abundant population of 293T cells, totaling to 1580 cells. Using a pre-processed and normalized expression containing 1000 filtered genes, we evaluated the performance of scAIDE (Figure 3A). According to the original publication and reproduced results, FiRE achieved an F1-score of 0.71 with 32 false positives. Using the clustering setting, we set $k = 2$ and obtained an F1-score of 1.0 for scAIDE (with default parameters), while SC3 achieved 0.94 with five false positives. CellSIUS also achieved an F1-score of 1.0, and GiniClust3 (parameter neighbors set to 10) achived a high score of 0.97 with two false positives. As depicted on the left panel of Figure 3A, five 293T cells (red) were well-mixed into the rare subpopulation of Jurkat cells (green). The AIDE embedding precisely delineates the subpopulation from the abundant group (right panel of Figure 3A). As mentioned in a previous study (24), clustering methods may perform poorly when the rare subpopulation percentage drops below 2%. We compared scAIDE with CellSIUS, GiniClust3 and FiRE for 0.5, 1, 1.5 and 2% mixtures of Jurkat cells (Supplementary Figure S10). scAIDE was able to separate the rare cell type (Jurkat cells) exactly in all four cases (Supplementary Figure S11).

To address the imbalanced composition of cell types, we further simulated rare cell type composition in datasets. We retained the largest two cell types and then sampled 50 or 500 (depending on the size of the dataset) cells for each of the remaining cell types. If the number of cells was less than the sampling number, we retained all cells of that particular cell type. RPH-kmeans is extensively compared with *k*-means++ and *k*-means (random initialization) to show its outstanding capability in detecting rare cell type subpopulations (Supplementary Figure S6, Note V and Tables S23–24). For both PCA and AIDE dimensional reduction techniques, RPH-kmeans (default parameters) outperformed the others in detecting small clusters (evaluated by ARI and NMI). Specifically, on PCA embeddings of the PBMC 68k dataset (Supplementary Figure S6a), RPH-kmeans achieved an ARI of 0.544 with only one initialization while traditional *k*-means++ algorithm only achieved

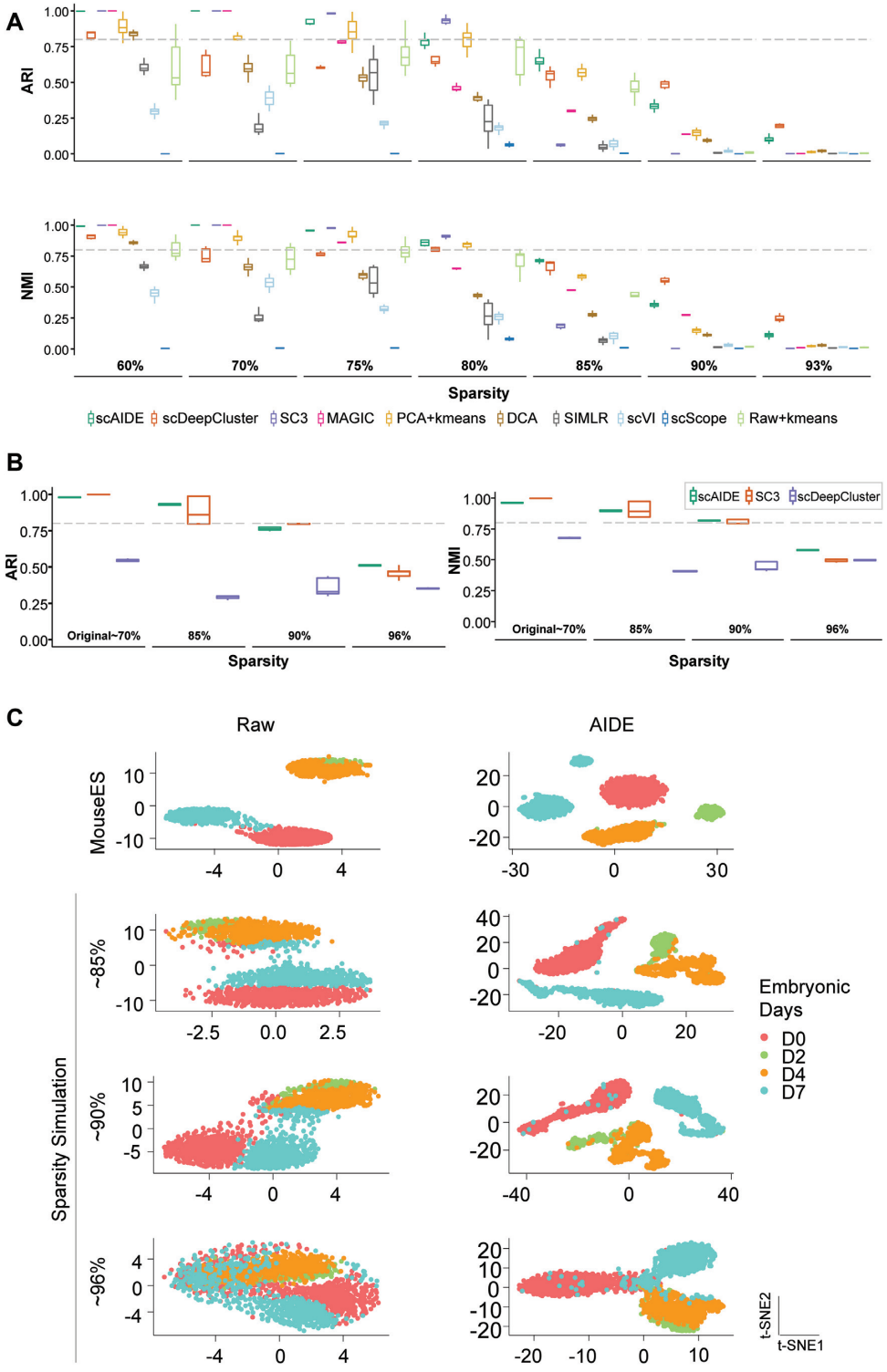

Figure 2. Dropout simulation and analysis. (**A**) Fully simulated single-cell datasets were generated using *splatter*, ranging from 60% sparsity to 93%. Boxplots follow similar settings to Figure 1B. We used the default parameters for scAIDE and set the maximum training steps to 40 000 without early stop. (**B**) Simulations were obtained by adding dropout events to the mouse ES dataset, increasing from 70% to about 96%. We used the default parameters for scAIDE. The left panel shows the ARI performance of scAIDE, SC3 and scDeepCluster, similarly for NMI on the right. The boxplots were drawn by running each algorithm five times and obtaining a distribution. (**C**) t-SNE visualizations of the raw gene expression matrix and the AIDE representation of the mouse ES dataset. Colors represent the true cell labels.

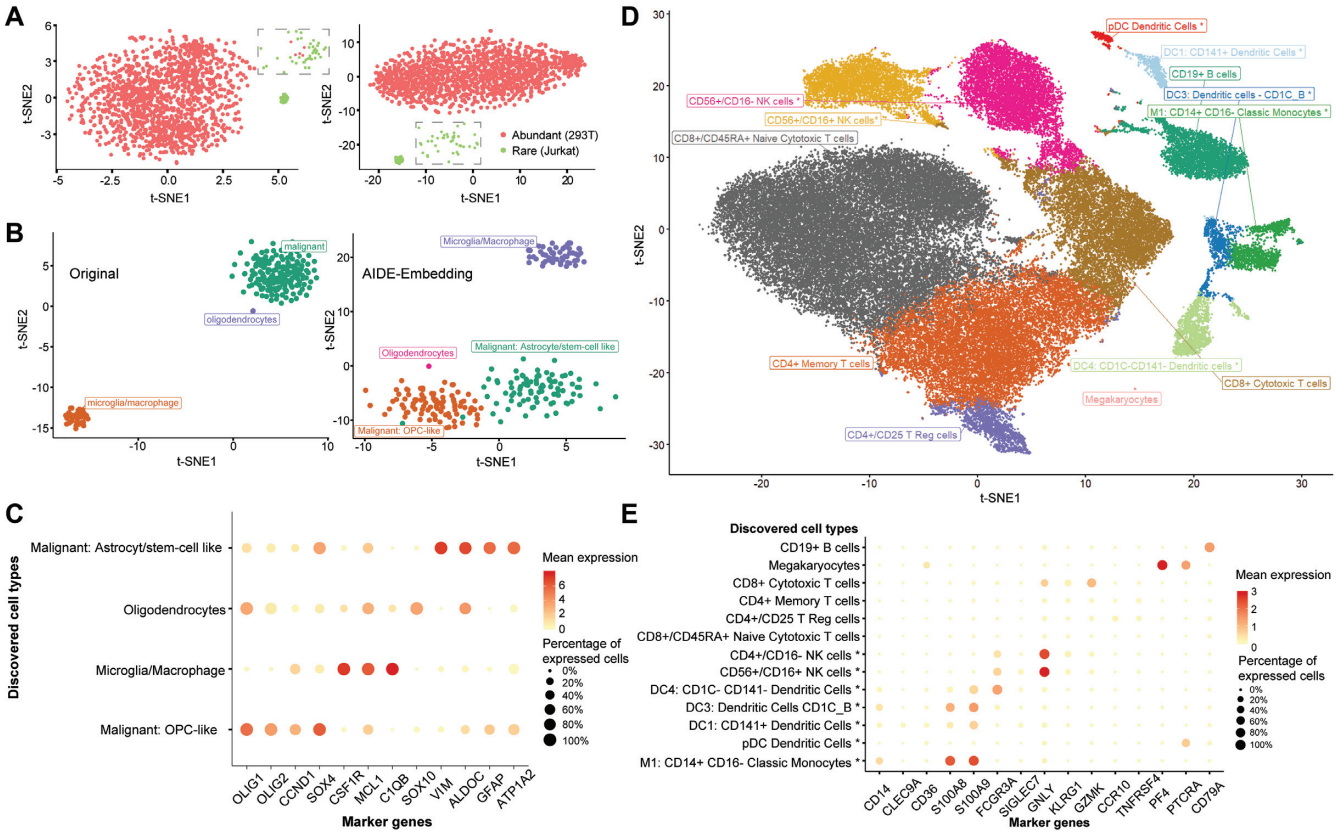

Figure 3. Deciphering cell subpopulations from global clustering results. (**A** and **B**) t-SNE plots showing the normalized gene expression profile with cell labels on the left, and t-SNE after AIDE embedding on the right with annotated labels determined from clustering results. (A) Simulated experiment on mixing 2.5% of the rare population (Jurkat cells in green) with an abundant population (293T cells in red). (B) A labeled glioblastoma dataset with 252 cells. OPC: oligodendrocyte progenitor cells. (**C**) Significantly expressed marker genes in the tumor dataset. The size of the points represents the percentage of cells that expressed the particular marker within the specific cell type. Color depicts the average log normalized expression. (**D** and **E**) Analysis of PBMC 68k dataset. Annotations with an asterisk (*) imply that they are new findings of cell subtypes. (D) t-SNE visualization of AIDE embedding with annotated labels. (E) Significantly expressed markers genes according to respective cell types within PBMC dataset.

0.343 (with 100 initializations). On the mouse retina dataset (Supplementary Figure S6b), where the composition of cell types is highly imbalanced, RPH-kmeans also achieved significantly better clustering results (ARI = 0.88 with 1 initialization) than k-means++ (ARI = 0.437 with 100 initializations). Thus, these results reflect that RPH-kmeans can identify rare subpopulations more accurately.

Case study I: Subpopulations of malignant cells in glioblastoma. Next, we applied scAIDE to a glioblastoma dataset (MGH107) (48). It is important to understand and dissect the underlying cell types in tumor micro-environments (TME). This dataset contains only 252 cells, and we were able to identify subpopulations in malignant cells (Figure 3B) using AIDE. As analyzed in one of our previous work (49), a subset of malignant cells express astrocytic and stem-cell-like genes (*GFAP*, *ALDOC*, *ATP1A2*, *VIM*), resembling the results shown in Figure 3B and C. Furthermore, we classified the second subset of malignant cells to be oligodendrocyte progenitor cell-like (OPC-like), expressing *OLIG1*, *OLIG2*, as well as proliferative markers such as *SOX4* and *CCND2*. Additionally, discovered markers can be found in Supplementary Table S7 (P-value $< 1 \times 10^{-10}$). The results demonstrate that malignant cells within TME have the potential to proliferate. By global unsupervised clustering analysis, we were able to identify a subgroup of malignant cells, adding to the understanding of development in tumors.

Case study II: Identification of common and rare subtypes in PBMCs. Subsequently, we applied our method to a larger dataset, containing about 68 000 PBMC cells (3). The original study reveals 11 cell types by calculating the correlation between reference transcriptomes to single-cell expressions. We first applied AIDE to generate a reduced embedding of the PBMC data (Figure 3D). We used weighted BIC (Methods) to automatically determine the number of clusters from the embedding ($k = 13$, Supplementary Figure S4a). Cluster-specific marker genes were cross-referenced with known markers (38) to annotate each cell type ('Materials and Methods' section). As a result, we identified subpopulations of natural killer (NK) cells, dendritic cells (DC), and monocytes. In particular, CD56+ CD16+ NK cell expresses *SIGLEC7* and *GNLY*, while CD56+ CD16- NK cell expresses *KLRG1* (50). Furthermore, four different subpopulations of DCs were identified as plasmacytoid DC (pDC), inflammatory DC, CD141+ DC and CD1C- CD141- DC. Some of their respective top markers are shown in Figure

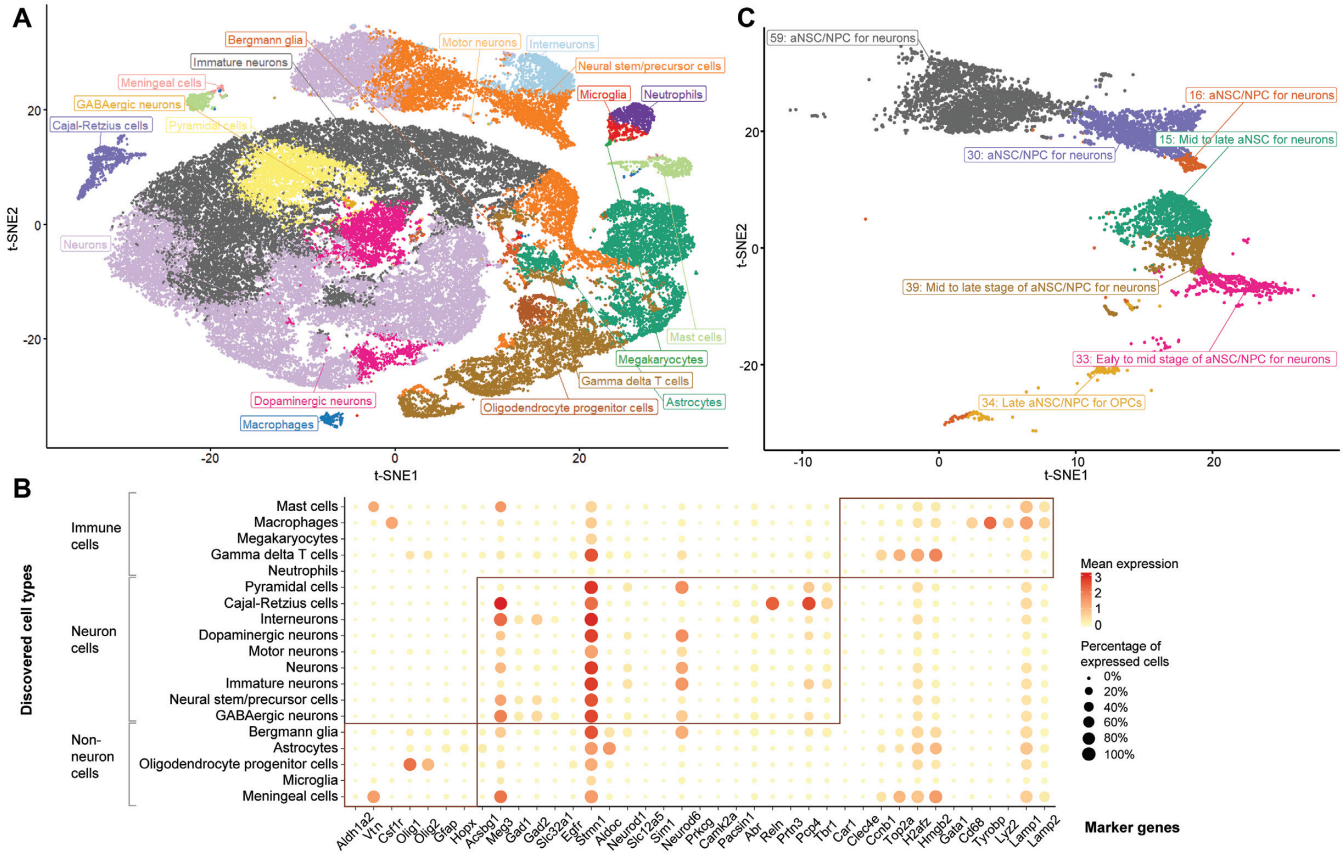

Figure 4. Putative and rare cell type discovery in the 1.3 million neural cell dataset. (**A**) t-SNE visualization of the AIDE embedding on 5% sampled data from the full dataset. Data are sampled based on the 64 clusters obtained from scAIDE. Colors represent the 19 cell type annotations from our biological analysis. (**B**) t-SNE visualization on cells annotated with the label: neural stem/precursor cells. These correspond to the orange cells in the same position as in A. Annotations were validated by different markers and their respective positioning. NPC: neural progenitor cells; aNSC: activated neural stem cells; OPC: oligodendrocyte progenitor cells. (**C**) Top significantly expressed marker genes in each respective cell types.

3E. As t-SNE components may not fully capture the global structure of data (51), we developed a simple approach (Methods) to visualize and identify possible development trajectory of cell clusters (Supplementary Figure S8). Interestingly, cell clusters of the lymphoid progenitor lineage (T cells, B cells and NK cells) are clearly separated from those of the myeloid progenitor lineage (dendritic cells, monocytes and megakaryocytes). Notably, we identified the rare cluster of megakaryocytes (about 0.24% of profiled cells) and the subpopulations of dendritic cells (ranged from 0.5 to 2.3% of the profiled population), similar to the results reported by the authors of FiRE (26).

Case study III: Cell type decomposition and primed differentiation process in a 1.3m neural dataset. Finally, to investigate the underlying biological insights in neural brain development, we applied scAIDE to provide a global clustering analysis on a 1.3 million neural dataset (27). Within 30 min, we obtained our embedding and 64 cluster assignments (Figure 4A and Supplementary Figure S4b). Using a curated list of markers from PanglaoDB (39), including both markers of neural and the immune system, we eventually mapped the 64 clusters to 19 putative cell types by their respective marker genes (Figure 4B). We further validated their cell types with an enrichment *P*-value (*P*-value < 0.05), shown in Supplementary Tables S8 and 9, except for the cluster of megakaryocytes which significantly expressed *Gata1*. Neuronal markers such as *Meg3* separates neuron cells from the rest of the cells. *Stmn1*, which had been reported to be highly expressed in later stages of neurogenesis (52), is consistently expressed throughout most neural cells (Figure 4B). Together, these results show that our clustering approach is capable of identifying putative cell types.

To further define cell subpopulations, we focus on the population of neural stem/precursor cells (orange cells shown in Figure 4A). A total of seven clusters were mapped to this cell type from global clustering (ranging from ∼0.28 to 2.4% of the total population). Hence, we attempted to delineate the different sub-types of neural stem cells or neural precursor cells. First, we visualized the separation of different cell clusters in a heatmap (Supplementary Figure S9) and identified three possible neural stem cell development based on AIDE embedding. Then, we utilized quiescent markers (*Clu*, *Id3*), activation markers (*Egfr*, *Atp1a2*, *Gfap*, *Prom1*), neurogenesis markers (*Dlx1*, *Dlx2, Dcx*, *Dlx6as1*) and astrocytic markers (*Fgfr3*, *Gja1*, *Jag1*) (53), to assign possible stages of cell types to each cluster (Figure 4C and Supplementary Tables S10–11). Interestingly, cluster 33 significantly expressed *Clu*, suggesting that a small portion of quiescent cells may be present. Since cluster 33 is clustered

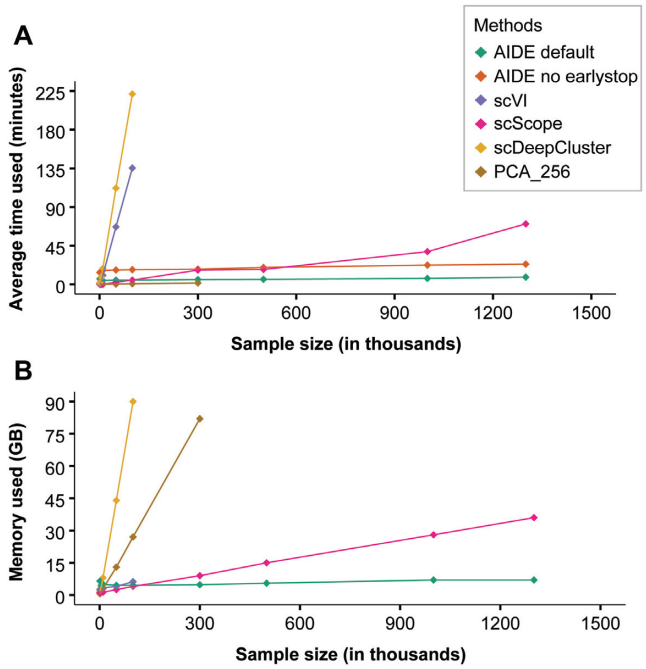

Figure 5. Comparison of computational efficiency. (**A** and **B**) Both graphs share the same legend, with (A) showing the computational time used in minutes and (B) depicting the amount of memory usage (in gigabytes, GB) for each method. As AIDE default and AIDE (no early stop) run with the same memory usage, we only plotted AIDE default in B.

with OPCs and astrocytes (Supplementary Figure S9), we believe that these are early- to mid-aNSCs primed for OPCs (significantly expressed in astrocytic markers with P-value < 0.05). Clusters 16, 30 and 59 distinctively express neurogenesis markers, suggesting a cell fate toward interneurons and neurons. For cluster 34, although it groups with gamma delta T cells, oligodendrocyte markers of *Pdgfra* and *Olig2* are significantly expressed, suggesting an OPC lineage. The last two groups of mid- to late-neural stem/progenitor cells (clusters 15 and 39) are grouped between gamma delta T cells and neuronal cells (Figure S9). Although gamma delta T cells respond to neuroinflammation, a recent study shows that they promote short-term memory by controlling synaptic plasticity in hippocampus neurons at steady-state (54). The organization of our identified cell clusters also suggests a supportive role of gamma delta T cells within neuronal cell types. In conclusion, we were able to define detailed cell subpopulations in single-cell datasets using the scAIDE clustering analysis framework.

Scalability

In Figure 5, we compared the scalability and efficiency of AIDE against the embedding step in current deep learning methods (except for scDeepCluster which directly generates the clustering result). Different sample sizes were sampled from the 1.3 million single-cell dataset to assess their performance, where the number of genes remains consistent. All experiments were performed on our CentOS system with 24 CPU cores at 2.5GHz, 125GB of memory and one 1080Ti graphics card. By default, AIDE reduces the data to 256 dimensions. The number of reduced dimensions were shown on the plot for all other methods. It should be noted that scScope pre-processed the data by selecting only the top 1000 variable genes as input to their algorithm. All other methods had the input of the full gene expression profile. Even with early stop disabled, we were able to obtain an AIDE embedding within 24 min using only 7 GB of memory on the full 1.3 million single-cell dataset.

Additionally, our clustering algorithm (RPH-kmeans) not only improves the detection of rare populations but is also more efficient than k-means++ or k-means (with random initialization) in some cases; where better-initialized centers lead to faster convergence and better result (Supplementary Figure S7). Thus, scAIDE is a highly efficient approach for analyzing huge single-cell datasets.

DISCUSSION

To date, it has been of great interest to use single-cell sequencing technology to identify both common and rare cell types in complex tissues. Most common cell types have been discovered long and are well-studied; however, rare subpopulations often remain obscure, particularly in diseases. Typical clustering approaches may be limited in both their ability to identify minor populations and computational time (55). We developed scAIDE to provide accurate and efficient clustering analysis, delineating both putative and rare cell types in single-cell datasets.

While many deep learning-based methods have been developed (17,19), including scalable methods such as scScope (46), we show that their performance is inconsistent between simulations and real datasets. Our analysis demonstrated the robustness of scAIDE in cases of high dropout rates and its ability to delineate rare cell types. In most cases, default parameters or small tuning would be sufficient.

In particular, we analyzed a small tumor dataset with scAIDE, identifying important sub-populations of malignant cells that express different lineage markers. Additionally, scAIDE was able to cluster rare cell types of megakaryocytes and dendritic subpopulations (ranging from 0.24 to 2.3% of profiled cells) in the PBMC 68k dataset. We also showed its ability to determine clusters which were correctly assigned to putative cell types (with significant enrichment p-values) in a 1.3 million neural cell dataset. Three different lineage development branches were identified by further investigation of seven clusters assigned to the neural stem/progenitor cells. Together, we demonstrate the capability of scAIDE to reveal putative and rare cell types in single-cell datasets. We believe that there is excellent potential for scAIDE to be further incorporated into trajectory development analysis in the future.

Within only 30 min, scAIDE could cluster the 1.3 million single-cell dataset using only 7 GB of memory. Together with its consistent performance and downstream biological analysis, we believe that our clustering analysis framework would provide a deepened understanding of cell types and developments within complex tissues and diseases.

DATA AVAILABILITY

scAIDE is publicly available via https://github.com/tinglabs/scAIDE.

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

ACKNOWLEDGEMENTS

K.X., Y.H., Z.F. and T.C. conceived and designed the study. Y.H. developed the computational method and evaluated the clustering performance. K.X. developed and performed the biological analysis. K.X., Y.H., Z.F. and Z.L. interpreted the results. K.X. and Y.H. wrote the manuscript. T.C. supervised the study. All authors have read and approved the submitted version.

FUNDING

National Natural Science Foundation of China [61872218, 61721003, 61673241, 61906105]; National Key R&D Program of China [2019YFB1404804]; Tsinghua-Fuzhou Institute of Digital Technology; Beijing National Research Center for Information Science and Technology (BNRist); Tsinghua University-Peking Union Medical College Hospital Initiative Scientific Research Program.

DECLARATION

The funders had no roles in study design, data collection and analysis, the decision to publish and preparation of the manuscript.
Conflict of interest statement. None declared.

REFERENCES

1. Tanay,A. and Regev,A. (2017) Scaling single-cell genomics from phenomenology to mechanism. *Nature*, **541**, 331–338.
2. Picelli,S., Faridani,O.R., Bjorklund,A.K., Winberg,G., Sagasser,S. and Sandberg,R. (2014) Full-length RNA-seq from single cells using Smart-seq2. *Nat. Protoc.*, **9**, 171–181.
3. Zheng,G.X., Terry,J.M., Belgrader,P., Ryvkin,P., Bent,Z.W., Wilson,R., Ziraldo,S.B., Wheeler,T.D., McDermott,G.P., Zhu,J. *et al.* (2017) Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.*, **8**, 1–12.
4. Rozenblatt-Rosen,O., Stubbington,M.J.T., Regev,A. and Teichmann,S.A. (2017) The Human Cell Atlas: from vision to reality. *Nature*, **550**, 451–453.
5. Pliner,H.A., Shendure,J. and Trapnell,C. (2019) Supervised classification enables rapid annotation of cell atlases. *Nat. Methods*, **16**, 983–986.
6. Wagner,F. and Yanai,I. (2018) Moana: a robust and scalable cell type classification framework for single-cell RNA-Seq data. bioRxiv doi: https://doi.org/10.1101/456129, 30 October 2018, pre-print: not peer-reviewed.
7. Kiselev,V.Y., Yiu,A. and Hemberg,M. (2018) scmap: projection of single-cell RNA-seq data across data sets. *Nat. Methods*, **15**, 359–362.
8. Abdelaal,T., Michielsen,L., Cats,D., Hoogduin,D., Mei,H., Reinders,M.J.T. and Mahfouz,A. (2019) A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biol.*, **20**, 194.
9. Kiselev,V.Y., Kirschner,K., Schaub,M.T., Andrews,T., Yiu,A., Chandra,T., Natarajan,K.N., Reik,W., Barahona,M., Green,A.R. *et al.* (2017) SC3: consensus clustering of single-cell RNA-seq data. *Nat. Methods*, **14**, 483–486.
10. Zurauskiene,J. and Yau,C. (2016) pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, **17**, 140.
11. Stuart,T., Butler,A., Hoffman,P., Hafemeister,C., Papalexi,E., Mauck,W.M. 3rd, Hao,Y., Stoeckius,M., Smibert,P. and Satija,R. (2019) Comprehensive integration of single-cell data. *Cell*, **177**, 1888–1902.
12. 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. *Nat. Methods*, **14**, 414–416.
13. Sinha,D., Kumar,A., Kumar,H., Bandyopadhyay,S. and Sengupta,D. (2018) dropClust: efficient clustering of ultra-large scRNA-seq data. *Nucleic Acids Res.*, **46**, e36.
14. Zeisel,A., Munoz-Manchado,A.B., Codeluppi,S., Lonnerberg,P., La Manno,G., Jureus,A., Marques,S., Munguba,H., He,L., Betsholtz,C. *et al.* (2015) Brain structure. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, **347**, 1138–1142.
15. Yang,L., Liu,J., Lu,Q., Riggs,A.D. and Wu,X. (2017) SAIC: an iterative clustering approach for analysis of single cell RNA-seq data. *BMC Genomics*, **18**, 689.
16. Hu,M.W., Kim,D.W., Liu,S., Zack,D.J., Blackshaw,S. and Qian,J. (2019) PanoView: an iterative clustering method for single-cell RNA sequencing data. *PLoS Comput. Biol.*, **15**, e1007040.
17. 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. *Nat. Commun.*, **10**, 390.
18. Risso,D., Perraudeau,F., Gribkova,S., Dudoit,S. and Vert,J.P. (2018) A general and flexible method for signal extraction from single-cell RNA-seq data. *Nat. Commun.*, **9**, 284.
19. Tian,T., Wan,J., Song,Q. and Wei,Z. (2019) Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nat. Mach. Intel.*, **1**, 191–198.
20. Pierson,E. and Yau,C. (2015) ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.*, **16**, 241.
21. Krzak,M., Raykov,Y., Boukouvalas,A., Cutillo,L. and Angelini,C. (2019) Benchmark and parameter sensitivity analysis of single-cell RNA-sequencing clustering methods. *Front. Genet.*, **10**, 1253.
22. Grun,D., Lyubimova,A., Kester,L., Wiebrands,K., Basak,O., Sasaki,N., Clevers,H. and van Oudenaarden,A. (2015) Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, **525**, 251–255.
23. Jiang,L., Chen,H., Pinello,L. and Yuan,G.C. (2016) GiniClust: detecting rare cell types from single-cell gene expression data with Gini index. *Genome Biol.*, **17**, 144.
24. Wegmann,R., Neri,M., Schuierer,S., Bilican,B., Hartkopf,H., Nigsch,F., Mapa,F., Waldt,A., Cuttat,R., Salick,M.R. *et al.* (2019) CellSIUS provides sensitive and specific detection of rare cell populations from complex single-cell RNA-seq data. *Genome Biol.*, **20**, 142.
25. Dong,R. and Yuan,G.C. (2020) GiniClust3: a fast and memory-efficient tool for rare cell type identification. *BMC Bioinformatics*, **21**, 158.
26. Jindal,A., Gupta,P., Jayadeva and Sengupta,D. (2018) Discovery of rare cells from voluminous single cell expression data. *Nat. Commun.*, **9**, 4719.
27. Genomics X. (2017) 1.3 million brain cells from E18 mice. https://support.10xgenomics.com/single-cell/datasets/1M_neurons.
28. Griveau,A., Borello,U., Causeret,F., Tissir,F., Boggetto,N., Karaz,S. and Pierani,A. (2010) A novel role for Dbx1-derived Cajal-Retzius cells in early regionalization of the cerebral cortical neuroepithelium. *PLoS Biol.*, **8**, e1000440.
29. Wolf,F.A., Angerer,P. and Theis,F.J. (2018) SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.*, **19**, 15.
30. Cox,T. and Cox,M. (2000) *Multidimensional Scaling*. 2nd edn. Chapman and Hall/CRC, Boca Raton.
31. Arthur,D. and Vassilvitskii,S. (2007) K-means++: The advantages of careful seeding. *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, pp. 1027–1035.
32. Gionis,A., Indyk,P. and Motwani,R. (1999) Similarity search in high dimensions via hashing. *Proceedings of the 25th International Conference on Very Large Data Bases*. Springer, pp. 518–529.
33. Datar,M., Indyk,P., Immorlica,N. and Mirrokni,V. (2004) Locality-sensitive hashing scheme based on p-stable distributions. *Proceedings of the Twentieth Annual Symposium on Computational Geometry*. Association for Computing Machinery, pp.253–262.
34. Jiang,L., Dong,Y., Chen,N. and Chen,T. (2017) DACE: a scalable DP-means algorithm for clustering extremely large sequence data. *Bioinformatics*, **33**, 834–842.

35. Li,Q., Wang,P., Wang,W., Hu,H., Li,Z. and Li,J. (2014) An efficient K-means clustering algorithm on MapReduce. *International Conference on Database Systems for Advanced Applications*. Springer, pp.357–371.
36. Hubert,L. and Arabie,P. (1985) Comparing partitions. *J. Classification*, **2**, 193–218.
37. Strehl,A. and Ghosh,J. (2003) Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. of Machine Learning Res.*, **3**, 583–617.
38. Villani,A.C., Satija,R., Reynolds,G., Sarkizova,S., Shekhar,K., Fletcher,J., Griesbeck,M., Butler,A., Zheng,S., Lazo,S. *et al.* (2017) Single-cell RNA-seq reveals new types of human blood dendritic cells, monocytes, and progenitors. *Science*, **356**, eaah4573.
39. Franzen,O., Gan,L.M. and Bjorkegren,J.L.M. (2019) PanglaoDB: a web server for exploration of mouse and human single-cell RNA sequencing data. *Database (Oxford)*, **2019**, baz046.
40. Zhou,Q., Chipperfield,H., Melton,D.A. and Wong,W.H. (2007) A gene regulatory network in mouse embryonic stem cells. *Proc. Natl. Acad. Sci. U.S.A.*, **104**, 16438–16443.
41. Zappia,L., Phipson,B. and Oshlack,A. (2017) Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.*, **18**, 174.
42. Deng,Q., Ramskold,D., Reinius,B. and Sandberg,R. (2014) Single-cell RNA-seq reveals dynamic, random monoallelic gene expression in mammalian cells. *Science*, **343**, 193–196.
43. Klein,A.M., Mazutis,L., Akartuna,I., Tallapragada,N., Veres,A., Li,V., Peshkin,L., Weitz,D.A. and Kirschner,M.W. (2015) Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**, 1187–1201.
44. van Dijk,D., Sharma,R., Nainys,J., Yim,K., Kathail,P., Carr,A.J., Burdziak,C., Moon,K.R., Chaffer,C.L., Pattabiraman,D. *et al.* Recovering gene interactions from single-cell data using data diffusion. *Cell*, **174**, 716–729.
45. Lopez,R., Regier,J., Cole,M.B., Jordan,M.I. and Yosef,N. (2018) Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, **15**, 1053–1058.
46. Deng,Y., Bao,F., Dai,Q., Wu,L.F. and Altschuler,S.J. (2019) Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning. *Nat. Methods*, **16**, 311–314.
47. Shekhar,K., Lapan,S.W., Whitney,I.E., Tran,N.M., Macosko,E.Z., Kowalczyk,M., Adiconis,X., Levin,J.Z., Nemesh,J., Goldman,M. *et al.* (2016) Comprehensive classification of retinal bipolar neurons by single-cell transcriptomics. *Cell*, **166**, 1308–1323.
48. Venteicher,A.S., Tirosh,I., Hebert,C., Yizhak,K., Neftel,C., Filbin,M.G., Hovestadt,V., Escalante,L.E., Shaw,M.L., Rodman,C. *et al.* (2017) Decoupling genetics, lineages, and microenvironment in IDH-mutant gliomas by single-cell RNA-seq. *Science*, **355**, 6332.
49. Xie,K., Liu,Z., Chen,N. and Chen,T. (2020) redPATH: reconstructing the pseudo development time of cell lineages in single-cell RNA-seq data and applications in cancer. bioRxiv doi: https://doi.org/10.1101/2020.03.05.977686, 06 March 2020, pre-print: not peer-reviewed.
50. Amand,M., Iserentant,G., Poli,A., Sleiman,M., Fievez,V., Sanchez,I.P., Sauvageot,N., Michel,T., Aouali,N., Janji,B. *et al.* (2017) Human CD56(dim)CD16(dim) cells as an individualized natural killer cell subset. *Front. Immunol.*, **8**, 699.
51. Kobak,D. and Berens,P. (2019) The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.*, **10**, 5416.
52. Shin,J., Berg,D.A., Zhu,Y., Shin,J.Y., Song,J., Bonaguidi,M.A., Enikolopov,G., Nauen,D.W., Christian,K.M., Ming,G.L. *et al.* (2015) Single-cell RNA-seq with waterfall reveals molecular cascades underlying adult neurogenesis. *Cell Stem Cell*, **17**, 360–372.
53. Dulken,B.W., Leeman,D.S., Boutet,S.C., Hebestreit,K. and Brunet,A. (2017) Single-cell transcriptomic analysis defines heterogeneity and transcriptional dynamics in the adult neural stem cell lineage. *Cell Rep.*, **18**, 777–790.
54. Ribeiro,M., Brigas,H.C., Temido-Ferreira,M., Pousinha,P.A., Regen,T., Santa,C., Coelho,J.E., Marques-Morgado,I., Valente,C.A., Omenetti,S. *et al.* (2019) Meningeal gammadelta T cell-derived IL-17 controls synaptic plasticity and short-term memory. *Sci. Immunol.*, **4**, eaay5199.
55. Campbell,J.N., Macosko,E.Z., Fenselau,H., Pers,T.H., Lyubetskaya,A., Tenen,D., Goldman,M., Verstegen,A.M., Resch,J.M., McCarroll,S.A. *et al.* (2017) A molecular census of arcuate hypothalamus and median eminence cell types. *Nat. Neurosci.*, **20**, 484–496.