scCobra: Contrastive cell embedding learning with domain-adaptation for single-cell data integration

Bowen Zhao[1], Dong-Qing Wei[1], Yi Xiong[1*], and Jun Ding[2*]

[1]State Key Laboratory of Microbial Metabolism, School of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai, 200240, China

[2]Meakins-Christie Laboratories, Department of Medicine, McGill University Health Centre, Montreal, QC, H4A 3J1, Canada

[*]All correspondences should be addressed to Y.X (xiongyi@sjtu.edu.cn) and J.D (jun.ding@mcgill.ca)

Abstract

The ever-increasing availability of single-cell transcriptomic data offers unrivaled opportunities to profile cellular states in various biological processes at high resolution, which has brought substantial advancements in understanding complex mechanisms underlying a large variety of bioprocesses. As limited by the protocol and technology, single-cell measurements in one study are often performed in batches, which unavoidably induces biological and technical differences in the single-cell measurements of the same study. Consequently, it presents challenges in analyzing all single-cell data from different batches together, particularly if the measurements were assayed with different technologies. Several methods have been developed to remove the aforementioned batch effects recently. However, there remain challenges unaddressed with those existing methods, including but not limited to the risk of over-correction, the need for the assumption of gene expression distribution, and expensive computation. To mitigate those limitations, we develop a novel deep learning method called scCobra that combines contrastive learning, domain adaptation, and generative adversarial networks to remove batch effects in single-cell RNA-seq data. The contrastive learning network is utilized to learn latent embeddings to represent the cells, domain-adaptation is employed to batch-normalize the latent embeddings of cells from distinct batches, while generative adversarial networks further optimize the blending effect. The proposed method does not require any prior assumption of gene expression distribution. We applied the scCobra method to one simulated and two real single-cell datasets with significantly experimental differences. Our method outperforms other benchmarked methods in batch correction and biological conservation, and its running efficiency is also among the best.

Keywords: Single-cell, Data integration, Batch correction, Contrastive learning, Domain adaptation, Generative adversarial networks

1 Introduction

Single-cell transcriptome technology substantially facilitates the explorations of complex and heterogeneous biological systems [1] with high-resolution cellular state measurements. In the past years, there have been ever-increasing single-cell maps of all different cell types in a large variety of organisms, such as humans, mice, and many other species[2-4]. However, the construction of these large-scale single-cell maps is generally conducted in multiple batches as limited by the current single-cell sequencing technology and other experimental considerations (e.g., budget or sample availability). In addition, differences in single-cell sequencing platforms, such as 10X genomics[5], micro-well[3], and smart-seq[6], also induce systematic technical variations between single-cell measurements in different batches, which is often termed as single-cell measurement "batch effects" [7-12]. To mitigate the impact of such batch effects on downstream single-cell data analytics, efficient computational methods to integrate single-cell data from different batches and remove the batch effects are essential. As of this date, dozens of single-cell data integration methods have been developed. The most widely used methods include SeuratV3[13], Scanorama[14], scVI[15], and Harmony[16]. SeuratV3[13] uses canonical correlation analysis to project the cells into a reduced latent space to identify correlations across datasets. The mutual nearest neighbors (MNNs) are then identified in the reduced latent space and serve as "anchors" to align the data from different batches. Another method Scanorama[14] also searches for MNNs in dimensionality-reduced space and uses them to guide batch correction. Unlike the SeuratV3 which searches similar cells across batch pairs to compute the correction, Scanorama searches across all batches and calculates the dataset merging priority based on the matching cell percentage in each batch. scVI [15] specifically models the gene expression variance induced by the library size difference and batch effects, which will be added as additional modules to a variational autoencoder that learns the cell embeddings by minimizing the cell gene expression reconstruction error, thereby separating biological signals from batch effects. An efficient method Harmony[16] first reduces the high-dimensional gene expression into the reduced space using principal component analysis (PCA)[17]. In the PCA reduced space, Harmony iteratively removes the present batch effects. In each iteration, the method clusters cells from different batches while maximizing the diversity of batches within each cluster and then calculates a correction factor for each cell to be integrated.

Although existing methods have demonstrated success in integrating single-cell datasets from different batches and even distinct sequencing platforms[16], various problems and limitations remain, leaving huge room for further improvement. For example, methods such as trVAE[18], Seurat V3[13], and MNN[9] still suffer from long running time and extensive memory usage, which potentially limits the application of those methods, particularly in large-scale single-cell datasets. Besides, a significant portion of existing methods, such as BBKNN[7], Scanorama[14], Seurat V3[13], and Harmony[16], are all based on MNNs. Those MNNs-based models find similar cells (MNNs) between batches and calculate the correction factor to mitigate the difference across batches. Therefore, this category of methods often over-corrects the batch effects since they tend to ignore the potential inherent difference associated with different batches, which could be problematic in many applications. Specifically, depending on specific experimental scenarios, the cellular states of even the same cell types across different experimental batches could differ, which is quite common in many disease studies, in which the same cell type (e.g., Macrophages) could be significantly different between healthy and disease patients[19]. Therefore, ignoring those inherent difference between batches may cripple the downstream single-cell data analytics. Deep learning-based methods such as scVI (auto-encoder based)[15] learn the reduced cell embeddings in the latent space by reconstructing the input gene expression. It often

assumes that the gene expression (i.e., raw count) follows a negative binomial distribution (or zero-inflated negative binomial), but this is still debatable [20] and it is also challenging to find a universal gene expression distribution for various single-cell datasets from different studies and by different sequencing platforms. A proper selection of the gene expression distribution will determine the reconstruction loss and dramatically influence the cell embedding learnings, which is critical for the downstream batch correction.

To overcome the above limitations, we propose a novel method called scCobra, which employs contrastive learning and domain adaptation to integrate single-cell data from different batches. We applied scCobra to simulated and real single-cell data from different batches and sequencing platforms. Our scCobra method demonstrated superior performance compared to several state-of-the-art methods. Furthermore, the running efficiency of scCobra is also among the best in all benchmarked methods, promoting its application in large-scale single-cell studies.

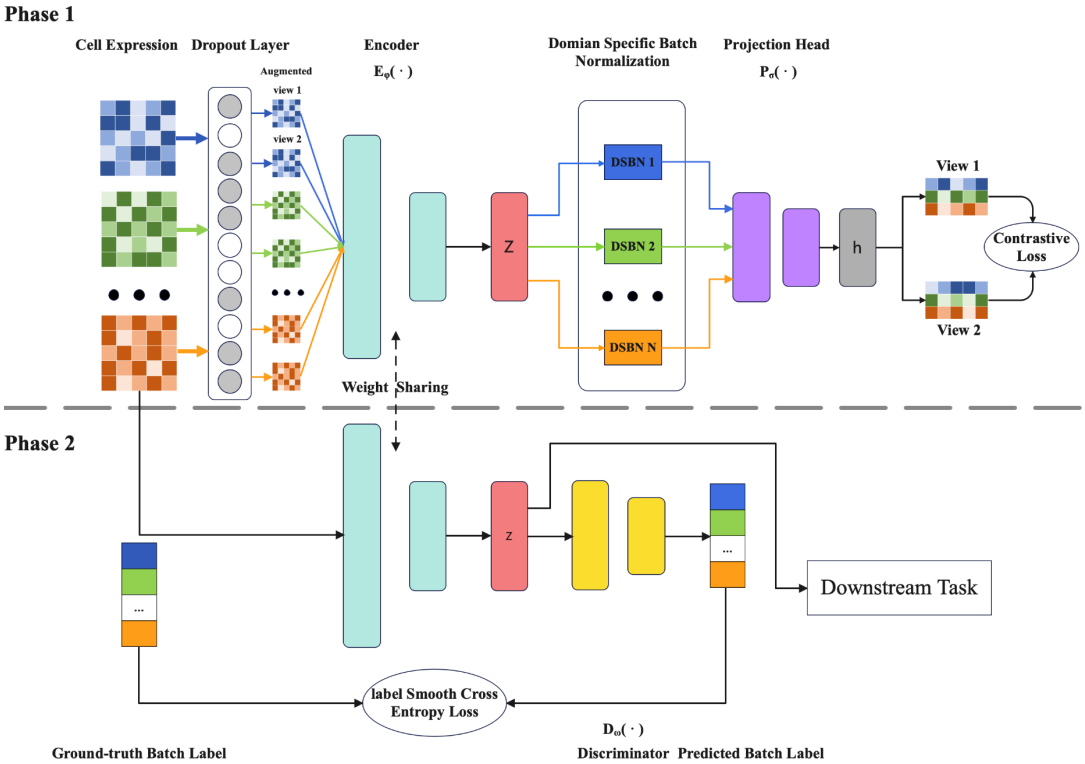

Fig.1 The scCobra method overview. scCobra is a two-phase model. In Phase 1, the method first employs contrastive learning and domain-specific batch normalization to learn cell embeddings with considering the batch effects in the data. For instance, the blue, green, and orange squares represent the datasets from different experimental batches to be integrated. Each cell is treated as input and sent to the random dropout layer twice to build two augmented views. Then, the encoder (Lake Blue) will encode the augmented views into embeddings, which will be then fed into the domain-specific batch normalization (DSBN) layer to further reduce into a lower dimensional space (***h***). The embedding outputs from the DSBN layer will be projected into a further reduced space for contrastive learning. In Phase 2, we employed a GAN to further optimize the data integration. The discriminator (yellow) will be built to predict the batch label of the embeddings (red). The cross-entropy discrimination loss will be backpropagated to optimize the encoder through adversarial training to remove the batch information from the cell embeddings. The final learned cell embeddings (z) will be used for downstream single-cell analytics.

2 Methods

2.1 Overview of scCobra

To resolve the single-cell data integration challenge and address the limitations of existing methods, here we proposed a computational model scCobra that leverages **co**ntrastive em**b**edding lea**r**ning with domain **a**daptation to integrate single-cell datasets from different batches. scCobra (Fig. 1) employs a two-step training strategy. In phase 1, we let the original data enter the random dropout layer twice independently, so each cell will get two augmented views. Then a weight-sharing encoder is employed to encode the cell inputs in two augmented views into reduced latent embeddings. Here we employ a set of domain-specific batch normalization layers to normalize the embeddings from various sources (experimental batches). Following that, the projection head maps the latent cell embeddings nonlinearly to the same joint reduced space ($\boldsymbol{h}$) for contrastive learning. In phase 2, we used the generative adversarial networks (GAN) to fuse different experiment batches of cells. Specifically, we use an adversarial training strategy to alternately train the discriminator and encoder, aligning the gene expression distribution of datasets from different experimental batches. After training, we use the trained encoder to encode cells directly, and the resulting cell embeddings (z) will be used for downstream tasks, such as cell clustering or trajectory inference of the integrated dataset. See the supplementary method for the specific parameters of scCobra.

2.2 Data Collection

In this study, we used one simulated and two actual single-cell datasets to evaluate our method, similar to the datasets that were used in the scIB's method[21]. These benchmarking datasets have already been preprocessed, normalized, and log1p transformed [22]. To improve the signal-to-noise ratio of those single-cell datasets, we selected highly variable genes to represent the cells in the preprocessed datasets using the scanpy[22] tool. Among them, the simulated dataset was generated by the Splatter package[23]. The dataset was composed of 12097 cells with 9979 genes, a mixture of seven cell types from six batches. The pancreas dataset was composed of 14 cell types mixed across nine sequencing platforms, where each sample of InDrop-seq data was regarded as an independent batch, including 16382 cells with 19093 genes. The pancreas dataset was collected from Gene Expression Omnibus (GEO) (GSE81076[24], GSE85241[25], GSE86469[25], GSE84133[26], GSE81608[26]). The third lung atlas dataset contains 16 batches with 17 cell types, including 32472 cells with 15148 genes, significantly larger and more complex than the other two datasets. This dataset is available with the GEO accession number GSE130148 [27] and the scIB publication [21].

2.3 Contrastive Learning Framework

Contrastive learning first emerged in computer vision. Methods such as Moco[28] and simCLR[29] were widely used to extract visual features to represent images in an unsupervised manner (i.e., embedding learning). Such a strategy can also be employed to learn reduced latent embeddings to represent the cells in single-cell data. In recent years, contrastive learning has been adopted by ever-increasing single-cell analytical methods. For example, Concerto[20] was developed for multimodal data integration, and contrastiveSC[30] was designed for clustering, which all employed contrastive learning. Contrastive learning is a paradigm where unlabeled data points are juxtaposed against each other to teach a model which points are similar and which are different. In single-cell studies, we will obtain multiple views of the same cell via data augmentation, which will then be contrasted to learn the reduced cell embeddings. Specifically, the contrastive neural network will project the high dimensional inputs into latent cell embeddings. The neural network is learned to maximize the similarity between positive cell pairs (augmented views of the same cells) while negative cell pairs (augmented views of different cells) will be pushed apart[31, 32]. Compared to other unsupervised cell embedding learning methods, such as variational autoencoder based scVI[15], the contrastive neural network does not

require an assumption for the gene expression distribution that is often challenging to specify. The contrastive cell embedding learning method is also running more efficiently because of the relatively simpler network structure.

Because of the aforementioned advantages, the contrastive neural network is preferred as a backbone model for single-cell data integration task in our study. Similar to the SimCSE[33] method, we employ dropout layers for data augmentations. Specifically, we will obtain two augmented gene expression matrices with a minimum difference from the same gene expression input matrix via one dropout layer twice. Formally, given a mini-batch cells of size N, scCobra performs dropout data augmentation on each cell x_i and results in $2N$ cells, where $d1$ and $d2$ represent two augmented views. For a specific cell x_i^{d1} in augmented view $d1$, there are $2N-1$ cell pairs in total (from intra and inter augmented views). The cell x_i^{d1} and its corresponding augmented cell x_i^{d2} in the other view were regarded as a positive pair $\{x_i^{d1}, x_i^{d2}\}$, and thus the remaining $2N-1$ cell pairs are all negative. Two data augment matrices are then fed into the deep neural net, composed of the encoder layer $E_{\varphi}(\cdot)$, domain-specific batch normalization layer $\mathrm{BN_j}(\cdot)$ (j represents the experimental batch), and projector layer $\mathrm{P}_{\sigma}(\cdot)$, which produces a pair of cell embeddings of 128 dimensions. Let $h_i^{d1}, \cdots, h_i^{d2}$ represent the output cell embeddings for contrastive learning. The pairwise similarity between cell embeddings can be measured by the cosine distance:

$$s\left(h_i^{d1}, h_j^{d2}\right) = \frac{\left(h_i^{d1}\right)\left(h_j^{d2}\right)^T}{\left\|h_i^{d1}\right\|\left\|h_j^{d2}\right\|} \tag{1}$$

Where s represents the cosine similarity, $d1, d2$represent the two dropout augmentation views, and $i, j \in [1, N]$ denotes the mini-batch with N cells. To optimize the above pairwise similarities, without loss of generality, the contrastive loss for a given cell x_i^{d1} can be written as:

$$l_i^{d1} = -log\frac{exp\left(s\left(h_i^{d1}, h_i^{d2}\right)/\tau\right)}{\sum_{j=1}^{N}\left[exp\left(s\left(h_i^{d1}, h_j^{d2}\right)/\tau\right) + exp\left(s\left(h_i^{d1}, h_j^{d2}\right)/\tau\right)\right]} \tag{2}$$

Where τ is the temperature parameter to control the sharpness. Then the total contrastive loss for all cells can be calculated as the following:

$$L_c = \frac{1}{2N}\sum_{i=1}^{N}\left(l_i^{d1} + l_i^{d2}\right) \tag{3}$$

2.4 Domain-specific Batch Normalization

Although contrastive learning provides an efficient and effective way to learn the cell embeddings, the model itself cannot remove batch effects from different single-cell datasets. Therefore, batch effect removal modules will be essential to be added to the backbone contrastive neural network. Domain adaptation has been commonly used for data integration[20, 34, 35]. Because of its advantage of matching the distribution of datasets from different sources without the need for manually annotated data, it could also be employed to integrate single-cell datasets of different batches. Domain adaptation is mainly to solve the difference between the distribution of the target domain and the source so that the model trained on the source can be transferred to the target data. Domain adaption often uses GAN for distribution matching and integrates data from different sources through adversarial training between generators and discriminators[36]. For instance, methods include iMAP[37], cellBlast[38], and scETM[39] remove single-cell batch effects through adversarial training.

Batch normalization, a commonly used normalization method in deep learning, could accelerate neural network convergence. Batchnorm1d denotes as:

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta \tag{4}$$

Where ϵ is a small number to prevent 0 in the denominator. γ and β are learnable parameters.

To be able to integrate single-cell datasets from different sources j (experiment batch), we add a specific normalization layer for each experimental batch of data between the encoder and projector in the contrastive learning framework. This domain-specific batch normalization layer (DSBN) is in the form of:

$$y_j = \frac{x_j - \mathrm{E}[x_j]}{\sqrt{\mathrm{Var}[x_j] + \epsilon}} * \gamma_j + \beta_j \tag{5}$$

With the DSBN layer, single-cell data from different sources are normalized independently, which integrates single-cell data from different batches.

2.5 Generative Adversarial Networks

To further improve the batch integration performance, we use generative adversarial networks (GAN) to remove the potential batch bias attached to the embedding z_i . First, we fix the parameters of the encoder $E_{\boldsymbol{\varphi}}(\cdot)$. We train a discriminator $D_{\omega}(\cdot)$ by minimizing the cross entropy[40] to predict the batch label b_j using latent embedding z_i learned from the encoder $E_{\boldsymbol{\varphi}}(\cdot)$. After training the discriminator, we fix the parameters of the discriminator $D_{\omega}(\cdot)$ and maximize the cross entropy so that the embedding encoded by the updated encoder $E_{\boldsymbol{\varphi}}(\cdot)$ can no longer be identified with the batch information b_j by the trained discriminator.

We then alternately train the encoder $E_{\boldsymbol{\varphi}}(\cdot)$ and discriminator $D_{\omega}(\cdot)$, and the training objective is defined as

$$\max_{\boldsymbol{\varphi}} \min_{\boldsymbol{\omega}} \sum_i CE\left(D_{\omega}\left(E_{\boldsymbol{\varphi}}(x_i)\right), b_j\right) \tag{6}$$

To avoid large batches distorting the discriminator's predictions, we use the label smooth cross entropy[41] instead of cross-entropy. The label smooth cross-entropy is written as:

$$\mathrm{Loss}_{LSCE} = \begin{cases} (1-\varepsilon)^{*}\mathrm{CE}, \text{ if } (y' = y) \\ \varepsilon^{*}\mathrm{CE} \text{ if } (y' \neq y) \end{cases} \tag{7}$$

Where Loss_{LSCE} represents the label smooth cross-entropy, ε represents the smooth parameter (default 0.1), CE represents the cross entropy, y' is the predicted experiment batch label, and y' is the true experiment batch label.

2.6 Batch Correction and Biological Conservation metrics

We use the Adjusted Rand Index (ARI)[42], Normalized Mutual Information (NMI)[43], and average silhouette width (ASW) across cell types[44] to measure the biological conservation of the data. The batch correction performance is evaluated by the following metrics: average silhouette width (ASW) across batches[44], k-nearest-neighbor batch effect test (kBET)[44] and integration Local Inverse Simpson's Index (iLISI)[16]. Please refer to the supplementary method for the detailed descriptions of all single-cell data integration evaluation metrics.

3 Results

scCobra can mitigate the batch effects while preserving the difference between cell types

We first applied our method scCobra to integrate a simulated benchmarking single-cell data integration dataset. For comparison purposes, we also applied a few state-of-art single-cell data integration methods that include Seurat, Scanorama, scVI, and Harmony to integrate the same simulated dataset [21]. The single-cell data integration results are shown in Fig 2. From the UMAP plots, we clearly observed that

the MNNs-based methods indeed tend to over-correct the batch effects, which negatively impact the downstream cell clustering and cell type identification tasks. For example, there are only six different cell types found after the batch effect correction by Seurat. Group 7 cells are indistinguishable from Group 2 cells in the batch effected data and thus were merged together, which is inconsistent with the ground-truth. On the other hand, the other MNNs-based method Harmony managed to capture all seven different cell groups. Nevertheless, it failed to remove the batch effects for many cells. For example, in Fig.2A, we can easily spot that there are multiple cell clusters dominated by cells from one batch. The scVI method splits the cells of Group 1 into 2 clusters, which indicates an insufficient batch effect removal as the smaller cluster is dominated by cells from batch 5. Scanorama and scCobra are among the best performing methods. They all captured 7 different cell groups (Fig 2B), and no significant batch effects were observed after batch correction. The biological conservation metrics comparison plot (Fig 2C) shows that the scCobra is competitive to the state-of-the-art scores, and the batch correction metrics comparison plot (Fig 2D) shows that scCobra outperformed all other methods in all metrics.

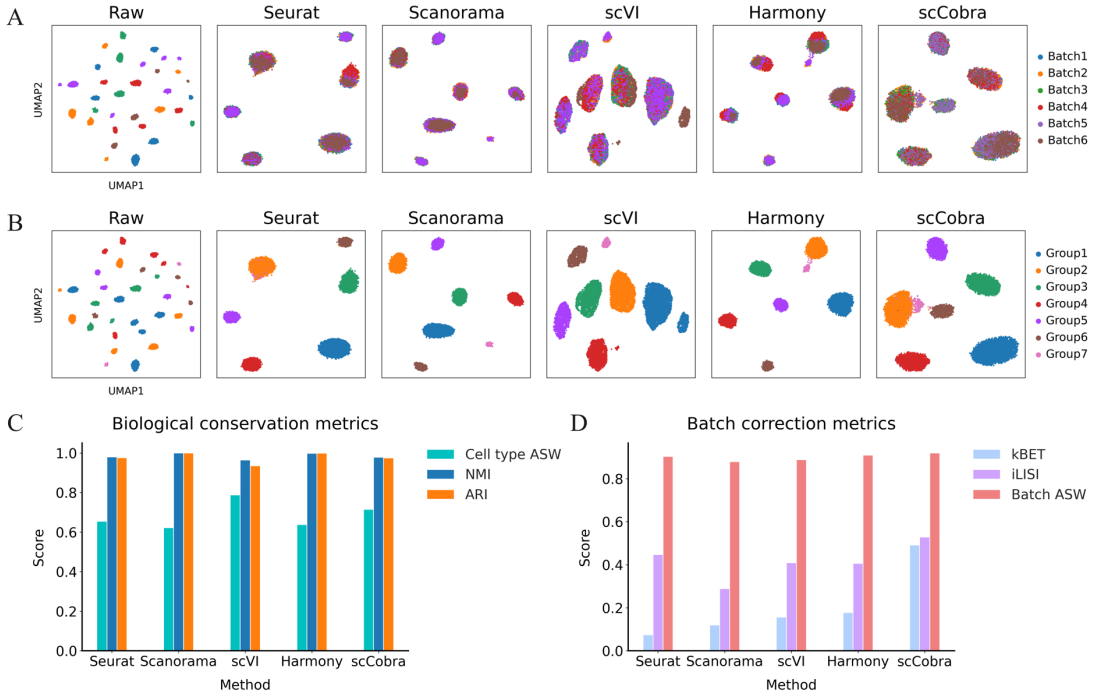

Fig. 2. scCobra outperforms all other benchmarked methods in integrating a simulated single-cell dataset. (A) shows the UMAP plots inferred from each benchmarked method. All the cells are color-coded by experimental batches. (B) shows the same UMAP plots color-coded by cell types/groups. (C, D) show all the benchmarking metrics of different methods.

scCobra can integrate single-cell data from multiple platforms

Differences in single-cell sequencing platforms will also introduce batch effects inevitably. To showcase the performance of scCobra in integrating single-cell datasets from different platforms, we evaluated whether scCobra can integrate single-cell data from different sequencing platforms on a pancreas single-cell dataset [21]. The pancreas dataset contains cells from six sequencing platforms (CEL-seq[45], CEL-seq2[46], Fluidigm c1[47], InDrop[48], SMARTer[49], and Smart-seq2[50]) and a total of nine batches of single-cell data. For benchmarking purposes, we applied the other 4 methods to the same pancreas single-cell data. The single-cell data integration results are shown in Fig 3.

The UMAP plots in Fig 3B show that Seurat could accurately distinguish most cell types and well integrate the single-cell dataset, but 'alpha' and 'acinar' cells are not accurately distinguished. Fig 3C and 3D show that Seraut's single-cell data integration performance is among the best on this dataset.

This implies that Seurat can perform well when potential over-correction is not a significant issue, like in this dataset when the cells from the same condition are sequenced by different platforms, and there is no strong biological inherent difference between batches, and thus we will not have to worry about the over-correction. The UMAP plot for Scanorama in Fig 3A implies that the method under-corrects all the cells from different sequencing platforms. Furthermore, Scanorama also cannot integrate 'alpha' cells, 'beta' cells, and 'gamma' cells from different sequencing platforms. The benchmarking metrics in Fig3. C and D clearly show its inferior performance in both biological conservation and batch correction. scVI single-cell data integration performance is also among the worst in all benchmarked methods. The UMAP plot in Fig.3A shows that scVI failed to integrate cells from most sequencing platforms, particularly cells from CEL-seq2 and Smart-seq2. The UMAP plot in Fig.3B suggests that different cell types are not clearly separated by the scVI method after the batch correction. The benchmarking metrics shown in Fig.3 C and 3D deliver the same message that scVI underperforms other methods on this dataset. Similar to the Seurat method, another MNNs-based method Harmony is also among the best in all benchmarked methods. Again, the potential over-correction risk associated with MNNs based method is minimized in this dataset (since the dataset is for the measurements of the same or similar set of cells from different platforms). Our method scCobra also performs well in this dataset. The UMAP plot in Fig 3A clearly supports that scCobra can integrate cells from different sequencing platforms well. The Fig.3B UMAP plot implies that the cell types were also well captured after scCobra batch correction. The benchmarking metrics in Fig.3 C and D demonstrate that the scCobra method is among the best on this dataset.

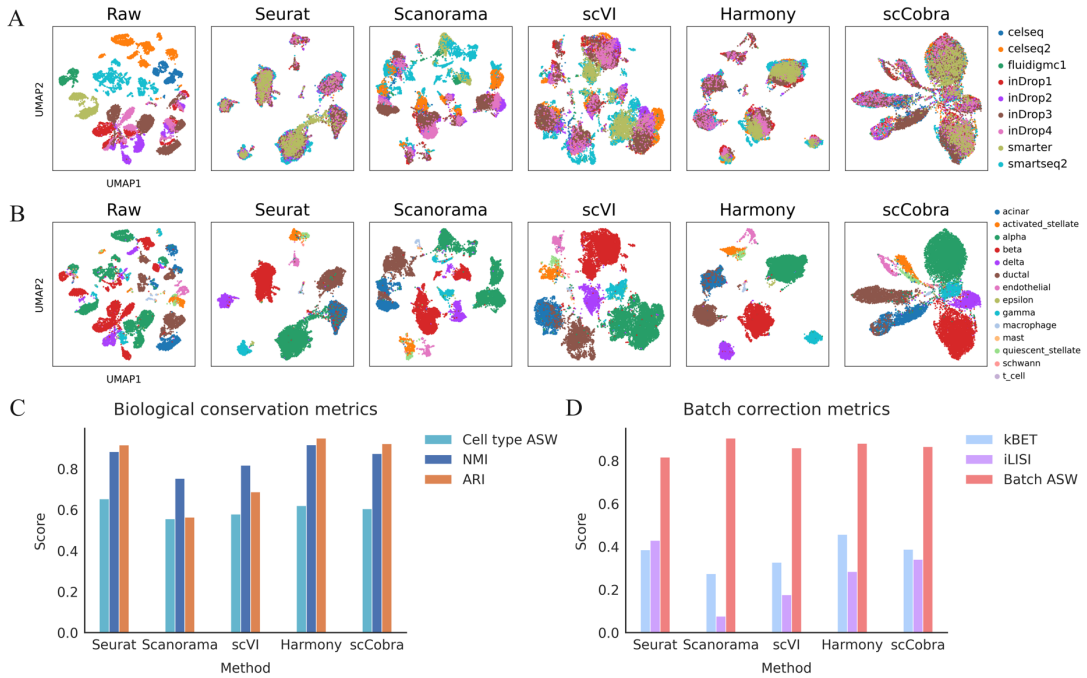

Fig. 3. scCobra can well integrate cells from different single-cell sequencing platforms. (A) shows the UMAP plots inferred from each benchmarked method. All the cells are color-coded by experimental batches. (B) shows the same UMAP plots color-coded by cell types/groups. (C, D) show all the benchmarking metrics of different methods.

scCobra enables the integration of complex single-cell data

The Lung atlas dataset[21] (32472 cells, 15148 genes) is a very complex dataset (in terms of 16 batches and 17 cell types). Again, scCobra is among the best performing method as benchmarked in this complex single-cell lung atlas dataset.

As shown in Fig. 4A UMAP plot, Seurat cannot integrate batch 6 and A2 well. In addition, different cell types after the batch correction by Seurat all mix, which again shows the risk of overcorrection of MNNs-based methods. This is also implied in the benchmarking metrics comparison (Fig4. C, D). Specifically, Seurat has one of the best batch correction metrics, but the biological conservation metrics are among the worst. The performance of Scanorama on this dataset is also unsatisfactory. The UMAP plots in Fig.4 A and B clearly show that Scanorama failed to batch-correct all the cells from different datasets and the separations between different cell types are unclear. The benchmarking metrics as shown in Fig4. C and D also show the overall performance of Scanorama is among the worst on this dataset. scVI better integrates single-cell data from multiple batches, but from the UMAP visualization in Fig 4B, scVI does not have clear cell type/group boundaries. Fig 4C and 4D show that scVI can best preserve biological information and remove batch effects. Its kBET score is the highest. Another MNNs-based method Harmony does not integrate batch B4 well (Fig.4A), but all the different cell types are not mixed after the batch correction. The evaluation metrics show an average performance of Harmony among all benchmarked methods. The performance of the other remaining 2 methods, scVI and scCobra, are close and among the best on this dataset. Each of the two best methods has its unique wins and losses. From Fig 4 A and B, scCobra well preserves biological information, and no significant batch effects were observed after correction. Fig.4 C and D demonstrate its superior performance over other methods. In general, the deep learning-based methods scVI and scCobra perform better on complex datasets. Nevertheless, the computing efficiency of scCobra is ~15x faster.

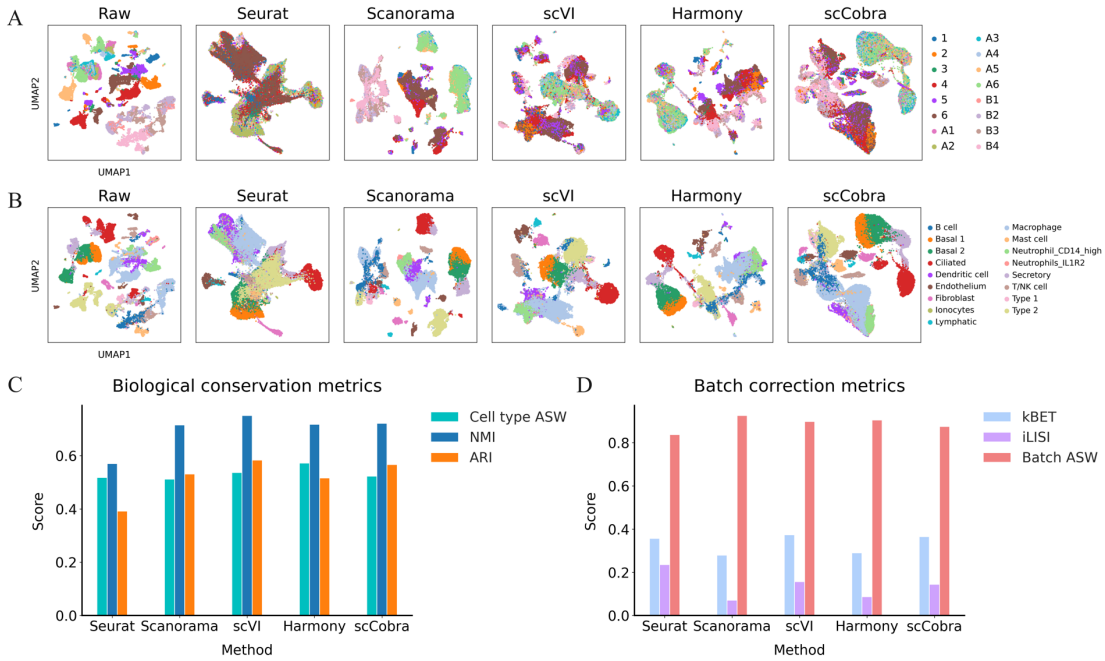

Fig. 4. scCobra performs well in integrating a complex single-cell dataset. (A) shows the UMAP plots inferred from each benchmarked method. All the cells are color-coded by experimental batches. (B) shows the same UMAP plots color-coded by cell types/groups. (C, D) show all the benchmarking metrics of different methods.

Domain-specific batch normalization and generative adversarial network are both indispensable for the superior batch correction performance

We further performed ablation experiments to demonstrate that all the modules in our scCobra method are essential, and removing any functional module will hamper the batch correction performance. We performed two groups of ablation on the single-cell pancreas dataset, namely scCobra without DSBN and scCobra without GAN. Single-cell data integration by scCobra without DSBN is significantly worse

than scCobra. The UMAP plots in Fig 5A show that all the batches cannot be well integrated by scCobra (w/o DSBN). Especially the InDrop3, almost no cells are integrated, and the UMAP plots in Fig 5B show that 'acinar' cells, 'alpha' cells and 'beta' cells were all segregated to some extent, indicating the negative impact of the batch correction on biological conservation. Fig.5 C shows that removing the DSBN module will worsen the biological conservation performance, while Fig.5 D implies that the batch correction metric will also be damaged if removing the DSBN module.

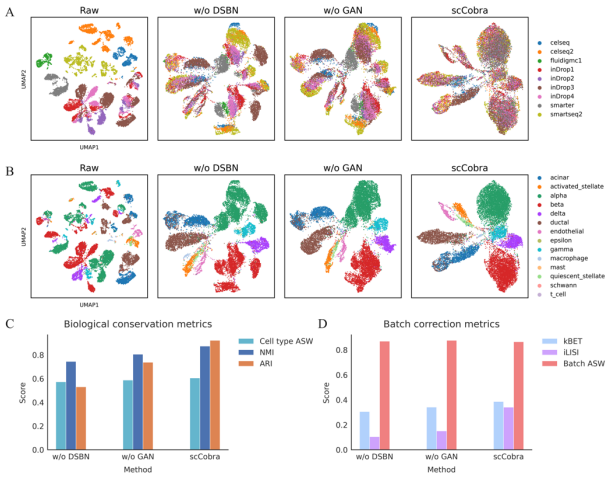

Fig. 5. DSBN and GAN modules are all indispensable for the superior performance of scCobra. (A) shows the UMAP plots inferred from each benchmarked method. All the cells are color-coded by experimental batches. (B) shows the same UMAP plots color-coded by cell types/groups. (C, D) show all the benchmarking metrics of different methods.

Revoking the GAN module also hurts the single-cell data integration performance. The UMAP plots in Fig 5A show that the batch mixing is not uniform, and the batch effects remain after the correction by scCobra (w/o GAN). The UMAP plots in Fig 5B show that many cell types are not well integrated (e.g., 'acinar' cells). Fig 5. C and D plots also indicate that removing the GAN module in scCobra will surely aggravate the single-cell dataset integration (in terms of both biological conservation and batch correction metrics).

The scCobra with both DSBN and GAN modules shows the best single-cell data integration performance. Through this ablation experiment, we can know that DSBN has a huge effect on the integration of single-cell batch data, but its integration could be uneven. The addition of the GAN module helps the model batch mix more uniformly, and cells of the same type are better clustered together. Therefore, the DSBN module and the GAN module are all indispensable for the superior single-cell data integration performance of scCobra.

scCobra is among the fastest single-cell data integration methods

We used the Intel Xeon ICX Platinum 8358 (32 cores) with 256GB memory to calculate and benchmark the runtime of Seurat, Scanorama, scVI, Harmony, and scCobra.The runtime comparison results are shown in Fig 6. Because we implemented the GAN module in the reduced space of the latent cell embeddings, GAN did not become a speed barrier for the model. In addition, studies have shown that the dual-tower structure of contrastive learning can greatly reduce time consumption[51]. Compared to other deep learning methods such as scVI, scCobra runs ~15x faster. Its running efficiency is very close to Harmony, which is known for its efficiency as it performs an iterative clustering optimization strategy that does not require heavy computation.

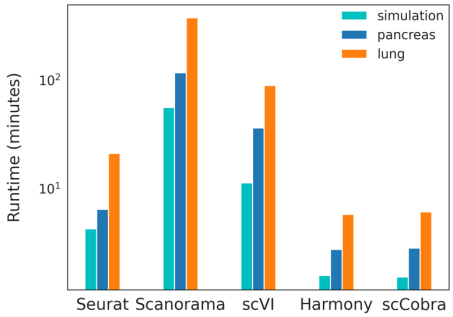

Fig.6. Runtime comparison of different single-cell data integration methods. We applied Seurat, Scanorama, scVI, Harmony, and scCobra to three single-cell datasets, including the simulated dataset, pancreas dataset, and lung atlas dataset. The bar represents the runtime in minutes, which is logarithmically scaled. Distinct colors are used to distinguish between different datasets.

All benchmarking metrics are detailed in the supplementary table S1-S5.

4 Discussion

In this study, we developed a novel single-cell data integration method that employs contrastive learning and domain adaptation, which outperforms other state-of-the-art methods in single-cell data integration in terms of multiple batch correction evaluation metrics. We applied scCobra in 3 different single-cell datasets that represent distinct single-cell data integration tasks, and the scCobra batch-corrected single-cell data well preserve the biological information while the batch effects are minimized (as documented by the biological conservation and batch correction metrics that we benchmarked).

The superior performance of the scCobra methods clearly originated from our novel strategy to piece together all different approved batch correction modules used in different fields (DSBN and GAN) to overcome the most significant limitations associated with existing methods: running efficiency, need for the prior assumption (e.g., gene expression distribution), and risk of over-correction. (1) The running efficiency of scCobra is the best among all benchmarked methods. Although the model is a deep neural network based approach and thus has a potential need for extensive computing resources for the training, we circumvented the problem by applying to the computationally expensive GAN module on the reduced latent embeddings. On the other hand, the DSBN module is very efficient by nature (as shown in the corresponding equation). A combination of both modules presents us with a powerful and efficient single-cell batch correction deep neural network. (2) scCobra does not require any assumptions about the data distribution. Compared with other deep neural network based methods such as scVI[15] and scANVI[52] that require a prior assumption for the gene expression distribution (e.g., negative binomial or zero-inflated negative binomial), our deep neural network based method scCobra with contrastive learning does not require any prior assumption for the gene expression, which is very beneficial as it is very challenging (if not infeasible) to find a universal gene expression distribution for all application scenarios. (3) scCobra is deep neural network based, which is different from many popular MNNs based batch correction methods (such as Seurat and Harmony). First, unlike many MNN methods, scCobra does not need to specify a batch as the anchor map, which avoids inducing bias if choosing an inappropriate anchor batch. Second, compared to the MNN-based methods, scCobra has a much lower overcorrection risk compared.

With the ablation experiments, we found that both DSBN and GAN modules are essential for superior single-cell data integration, which also supports that the novel marriage of these two batch correction strategies as implemented in our scCobra method indeed brings substantial benefits to the single-cell batch correction task. Although our method is very efficient and presents satisfactory single-cell integration, current scCobra can only work for single-cell RNA-seq data. Theoretically, scCobra, with proper extensions and modifications, could be used to integrate single-cell data from other omics and even single-cell multi-omics data that simultaneously profiles multiple cell modalities. However, this is beyond the scope of this study, and we hope to extend our scCobra to integrate single-cell multi-omics data in the near future.

5 Data and Code availability

The data we used and the implementation of scCobra are available at https://github.com/GlancerZ/scCobra.

6. Acknowledgements

This work was partially supported by CIHR PJT-180505, FRQS 295298, 295299, and NSERC RGPIN-2022-04399 to J.D. Y.X is supported by grants from the National Science Foundation of China (62172274).

References

1. Shalek, A.K., Satija, R., Shuga, J., Trombetta, J.J., Gennert, D., Lu, D., Chen, P., Gertner, R.S., Gaublomme, J.T., Yosef, N., Schwartz, S., Fowler, B., Weaver, S., Wang, J., Wang, X., Ding, R., Raychowdhury, R., Friedman, N., Hacohen, N., Park, H., May, A.P., Regev, A.: Single-cell RNA-seq reveals dynamic paracrine control of cellular variation. Nature 510, 363-369 (2014)

2. Han, X., Zhou, Z., Fei, L., Sun, H., Wang, R., Chen, Y., Chen, H., Wang, J., Tang, H., Ge, W.: Construction of a human cell landscape at single-cell level. Nature 581, 303-309 (2020)

3. Han, X., Wang, R., Zhou, Y., Fei, L., Sun, H., Lai, S., Saadatpour, A., Zhou, Z., Chen, H., Ye, F.: Mapping the mouse cell atlas by microwell-seq. Cell 172, 1091-1107. e1017 (2018)

4. Shahan, R., Hsu, C.-W., Nolan, T.M., Cole, B.J., Taylor, I.W., Greenstreet, L., Zhang, S., Afanassiev, A., Vlot, A.H.C., Schiebinger, G.: A single-cell Arabidopsis root atlas reveals developmental trajectories in wild-type and cell identity mutants. Developmental cell 57, 543-560. e549 (2022)

5. Macosko, E.Z., Basu, A., Satija, R., Nemesh, J., Shekhar, K., Goldman, M., Tirosh, I., Bialas, A.R., Kamitaki, N., Martersteck, E.M., Trombetta, J.J., Weitz, D.A., Sanes, J.R., Shalek, A.K., Regev, A., McCarroll, S.A.: Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. Cell 161, 1202-1214 (2015)

6. Picelli, S.: Full-Length Single-Cell RNA Sequencing with Smart-seq2. Methods Mol Biol 1979, 25-44 (2019)

7. Polański, K., Young, M.D., Miao, Z., Meyer, K.B., Teichmann, S.A., Park, J.-E.: BBKNN: fast batch alignment of single cell transcriptomes. Bioinformatics 36, 964-965 (2020)

8. Amodio, M., Van Dijk, D., Srinivasan, K., Chen, W.S., Mohsen, H., Moon, K.R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M.: Exploring single-cell data with deep multitasking neural networks. Nature methods 16, 1139-1145 (2019)

9. Haghverdi, L., Lun, A.T., Morgan, M.D., Marioni, J.C.: Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nature biotechnology 36, 421-427 (2018)

10. Li, X., Wang, K., Lyu, Y., Pan, H., Zhang, J., Stambolian, D., Susztak, K., Reilly, M.P., Hu, G., Li, M.: Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. Nature communications 11, 1-14 (2020)

11. Lotfollahi, M., Wolf, F.A., Theis, F.J.: scGen predicts single-cell perturbation responses. Nature methods 16, 715-721 (2019)

12. Welch, J.D., Kozareva, V., Ferreira, A., Vanderburg, C., Martin, C., Macosko, E.Z.: Single-cell multi-omic integration compares and contrasts features of brain cell identity. Cell 177, 1873-1887. e1817 (2019)

13. Hao, Y., Hao, S., Andersen-Nissen, E., Mauck, W.M., III, Zheng, S., Butler, A., Lee, M.J., Wilk, A.J., Darby, C., Zager, M., Hoffman, P., Stoeckius, M., Papalexi, E., Mimitou, E.P., Jain, J., Srivastava, A., Stuart, T., Fleming, L.M., Yeung, B., Rogers, A.J., McElrath, J.M., Blish, C.A., Gottardo, R., Smibert, P., Satija, R.: Integrated analysis of multimodal single-cell data. Cell 184, 3573-3587.e3529 (2021)

14. Hie, B., Bryson, B., Berger, B.: Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nature Biotechnology 37, 685-691 (2019)

15. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I., Yosef, N.: Deep generative modeling for single-cell transcriptomics. Nature Methods 15, 1053-1058 (2018)

16. Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.-r., Raychaudhuri, S.: Fast, sensitive and accurate integration of single-cell data with

Harmony. Nature methods 16, 1289-1296 (2019)
17. Lever, J., Krzywinski, M., Altman, N.: Principal component analysis. Nature Methods 14, 641-642 (2017)
18. Lotfollahi, M., Naghipourfar, M., Theis, F.J., Wolf, F.A.: Conditional out-of-distribution generation for unpaired data using transfer VAE. Bioinformatics 36, i610-i617 (2020)
19. Yuan, H., Yan, M., Zhang, G., Liu, W., Deng, C., Liao, G., Xu, L., Luo, T., Yan, H., Long, Z., Shi, A., Zhao, T., Xiao, Y., Li, X.: CancerSEA: a cancer single-cell state atlas. Nucleic Acids Research 47, D900-D908 (2018)
20. Yang, M., Yang, Y., Xie, C., Ni, M., Liu, J., Yang, H., Mu, F., Wang, J.: Contrastive learning enables rapid mapping to multimodal single-cell atlas of multimillion scale. Nature Machine Intelligence 4, 696-709 (2022)
21. Luecken, M.D., Büttner, M., Chaichoompu, K., Danese, A., Interlandi, M., Mueller, M.F., Strobl, D.C., Zappia, L., Dugas, M., Colomé-Tatché, M., Theis, F.J.: Benchmarking atlas-level data integration in single-cell genomics. Nature Methods 19, 41-50 (2022)
22. Wolf, F.A., Angerer, P., Theis, F.J.: SCANPY: large-scale single-cell gene expression data analysis. Genome biology 19, 1-5 (2018)
23. Zappia, L., Phipson, B., Oshlack, A.: Splatter: simulation of single-cell RNA sequencing data. Genome biology 18, 1-15 (2017)
24. Grün, D., Muraro, M.J., Boisset, J.-C., Wiebrands, K., Lyubimova, A., Dharmadhikari, G., van den Born, M., Van Es, J., Jansen, E., Clevers, H.: De novo prediction of stem cell identity using single-cell transcriptome data. Cell stem cell 19, 266-277 (2016)
25. Muraro, M.J., Dharmadhikari, G., Grün, D., Groen, N., Dielen, T., Jansen, E., Van Gurp, L., Engelse, M.A., Carlotti, F., De Koning, E.J.: A single-cell transcriptome atlas of the human pancreas. Cell systems 3, 385-394. e383 (2016)
26. Baron, M., Veres, A., Wolock, S.L., Faust, A.L., Gaujoux, R., Vetere, A., Ryu, J.H., Wagner, B.K., Shen-Orr, S.S., Klein, A.M.: A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. Cell systems 3, 346-360. e344 (2016)
27. Vieira Braga, F.A., Kar, G., Berg, M., Carpaij, O.A., Polanski, K., Simon, L.M., Brouwer, S., Gomes, T., Hesse, L., Jiang, J.: A cellular census of human lungs identifies novel cell states in health and in asthma. Nature medicine 25, 1153-1163 (2019)
28. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum Contrast for Unsupervised Visual Representation Learning. pp. arXiv:1911.05722 (2019)
29. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning, pp. 1597-1607. PMLR, (Year)
30. Ciortan, M., Defrance, M.: Contrastive self-supervised clustering of scRNA-seq data. BMC Bioinformatics 22, 280 (2021)
31. Wang, T., Isola, P.: Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. pp. arXiv:2005.10242 (2020)
32. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big Self-Supervised Models are Strong Semi-Supervised Learners. pp. arXiv:2006.10029 (2020)
33. Gao, T., Yao, X., Chen, D.: SimCSE: Simple Contrastive Learning of Sentence Embeddings. pp. arXiv:2104.08821 (2021)
34. Seo, S., Suh, Y., Kim, D., Kim, G., Han, J., Han, B.: Learning to optimize domain specific normalization for domain generalization. In: European Conference on Computer Vision, pp. 68-83.

Springer, (Year)
35. Wang, Y., Huang, W., Sun, F., Xu, T., Rong, Y., Huang, J.: Deep multimodal fusion by channel exchanging. Advances in Neural Information Processing Systems 33, 4835-4845 (2020)
36. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM 63, 139-144 (2020)
37. Wang, D., Hou, S., Zhang, L., Wang, X., Liu, B., Zhang, Z.: iMAP: integration of multiple single-cell datasets by adversarial paired transfer networks. Genome Biology 22, 63 (2021)
38. Cao, Z.-J., Wei, L., Lu, S., Yang, D.-C., Gao, G.: Searching large-scale scRNA-seq databases via unbiased cell embedding with Cell BLAST. Nature Communications 11, 3458 (2020)
39. Zhao, Y., Cai, H., Zhang, Z., Tang, J., Li, Y.: Learning interpretable cellular and gene signature embeddings from single-cell transcriptomic data. Nature Communications 12, 5261 (2021)
40. De Boer, P.-T., Kroese, D.P., Mannor, S., Rubinstein, R.Y.: A tutorial on the cross-entropy method. Annals of operations research 134, 19-67 (2005)
41. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818-2826. (Year)
42. Hubert, L., Arabie, P.: Comparing partitions. Journal of classification 2, 193-218 (1985)
43. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V.: Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12, 2825-2830 (2011)
44. Büttner, M., Miao, Z., Wolf, F.A., Teichmann, S.A., Theis, F.J.: A test metric for assessing single-cell RNA-seq batch correction. Nature methods 16, 43-49 (2019)
45. Hashimshony, T., Wagner, F., Sher, N., Yanai, I.: CEL-Seq: single-cell RNA-Seq by multiplexed linear amplification. Cell reports 2, 666-673 (2012)
46. Hashimshony, T., Senderovich, N., Avital, G., Klochendler, A., De Leeuw, Y., Anavy, L., Gennert, D., Li, S., Livak, K.J., Rozenblatt-Rosen, O.: CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq. Genome biology 17, 1-7 (2016)
47. Islam, S., Zeisel, A., Joost, S., La Manno, G., Zajac, P., Kasper, M., Lönnerberg, P., Linnarsson, S.: Quantitative single-cell RNA-seq with unique molecular identifiers. Nature methods 11, 163-166 (2014)
48. Klein, A.M., Mazutis, L., Akartuna, I., Tallapragada, N., Veres, A., Li, V., Peshkin, L., Weitz, D.A., Kirschner, M.W.: Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell 161, 1187-1201 (2015)
49. Verboom, K., Everaert, C., Bolduc, N., Livak, K.J., Yigit, N., Rombaut, D., Anckaert, J., Lee, S., Venø, M.T., Kjems, J.: SMARTer single cell total RNA sequencing. Nucleic Acids Research 47, e93-e93 (2019)
50. Picelli, S., Faridani, O.R., Björklund, Å.K., Winberg, G., Sagasser, S., Sandberg, R.: Full-length RNA-seq from single cells using Smart-seq2. Nature protocols 9, 171-181 (2014)
51. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. arXiv preprint arXiv:1908.10084 (2019)
52. Xu, C., Lopez, R., Mehlman, E., Regier, J., Jordan, M.I., Yosef, N.: Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. Molecular systems biology 17, e9620 (2021)