CSIAM Trans. Life Sci.
doi: 10.4208/csiam-ls.SO-2025-0025

Vol. **2**, No. 1, pp. 177-202
March 2026

Effectively Preserving Biological Variations in Multi-Batch and Multi-Condition Single-Cell Data Integration

Qingbin Zhou[1], Tao Ren[2,3], Fan Yuan[4], Jiating Yu[5], Jiacheng Leng[6], Jiahao Song[1], Duanchen Sun[1,7,*] and Ling-Yun Wu[2,3,*]

[1] *School of Mathematics, Shandong University, Jinan 250100, China.*
[2] *State Key Laboratory of Mathematical Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.*
[3] *School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China.*
[4] *School of Mathematics and Information Science, Yantai University, Yantai 264005, China.*
[5] *School of Mathematics and Statistics, Nanjing University of Information Science & Technology, Nanjing 210044, China.*
[6] *Zhejiang Lab, Hangzhou 311121, China.*
[7] *Shandong Key Laboratory of Cancer Digital Medicine, Jinan 250033, China.*

Received 13 August 2025; Accepted 17 October 2025

Abstract. Understanding phenotypic differences at the cell level is critical for comprehending the underlying pathogenesis of related complex diseases. However, the biological variations are obscured by batch effects, posing a challenge for integrating multi-batch and multi-condition single-cell datasets. Here, we present scFLASH, a deep learning-based model specially designed to explore single-cell biological variations while correcting undesired batch effects. scFLASH employs a conditional variational autoencoder with adversarial training to separate biological variations from technical noise and introduces a penalized condition classifier to preserve condition-specific biological signals. Through comprehensive benchmarking evaluations, scFLASH shows superior integration performances compared to other state-of-the-art methods. Applied to datasets such as Alzheimer's disease, COVID-19, and diabetes, we demonstrate that scFLASH is applicable to various scenarios, effectively integrating datasets with two or more conditions and different batch sources. scFLASH can enhance the gene expression profiles and identify the condition-related cell subpopulations, facilitating downstream analyses and offering biological insights into the cellular mechanisms of disease pathology.

AMS subject classifications: 92D10, 92-08, 68T05

Key words: Biological variations, data integration, batch correction, deep learning.

*Corresponding author. *Email addresses:* dcsun@sdu.edu.cn (D. Sun), lywu@amss.ac.cn (L.-Y. Wu)

1 Background

Single-cell RNA sequencing (scRNA-seq) has become an indispensable tool for understanding cellular heterogeneity and functionality, paving the way for personalized medicine and drug development [21, 28, 44, 52]. Among the single-cell analyses, deciphering phenotypic differences at the single-cell level is critical for uncovering the underlying pathogenesis of related complex diseases [4, 40, 55]. One practical approach to achieving this goal is integrating scRNA-seq from diverse sources and conditions [56,57]. However, the biological variations across phenotypes are entangled with batch effects, posing a challenge for integrating multi-batch and multi-condition (MBMC) single-cell datasets [6,29].

Several MBMC integration methods have recently been developed to remove technical variations while preserving phenotypic differences. scINSIGHT [37] utilized nonnegative matrix factorization to learn common and specific gene expression modules under different conditions. It employed linear assumptions to expression decomposition, which may hinder its model from effectively analyzing the complex nonlinear data. scMerge2 [24] used factor analysis to remove unwanted technical variations, whereas it only introduced pseudo-replicates within each condition and did not consider necessary connections between different conditions. Besides, scMerge2 also requires specific model inputs and thus cannot flexibly handle a single dataset under the same conditions. scDisInFact [56] was a deep learning-based model that used multiple encoders to disentangle gene expressions into shared and un-shared biological factors. Its established independent encoders may overlook the interactions between condition-specific biological signals. scDisco [25] was a recently proposed approach that employed condition and domain-specific layers in variational autoencoders to enhance the condition representations.

In addition to the inherent limitations of the methods discussed above, which affect their utility and versatility, the MBMC integration problem faces several key challenges. First, differences between conditions and cellular heterogeneity are both necessary biological variations that should be preserved. It is important to decipher condition differences without sacrificing the accurate representation of cell type heterogeneity characterization. Second, overly emphasizing batch correction can inadvertently distort or suppress biological variations across conditions, hindering the extraction of meaningful biological insights. Overcorrection should be avoided to prevent excessive data homogenization and misleading downstream analyses. Last, an advanced MBMC integration algorithm should be appropriate for various scenarios, effectively handling tasks with multiple conditions and diverse batch sources.

To this end, we developed a deep learning-based model, scFLASH, to explore phenotypic differences in MBMC integration. scFLASH utilizes a conditional variational autoencoder (CVAE) [48] framework to learn a disentangled latent space, capturing embeddings for known conditional attributes and unknown biological attributes. To remove batch effects, scFLASH applies adversarial training in the latent space. Furthermore,

scFLASH incorporates a penalized condition classifier that preserves condition-specific differences. We showed that scFLASH has superior integration performances compared to other state-of-the-art methods and enables a more balanced integration for preserving biological variations and removing technical noise. In real applications, scFLASH can effectively integrate scRNA-seq datasets with two or more conditions and different batch sources. Our studies demonstrated that scFLASH can facilitate downstream analyses by enhancing gene expression profiles and identifying condition-related cell subpopulations, providing biological insights into the cellular mechanisms of disease pathology.

2 Result

2.1 Overview of scFLASH

scFLASH is a deep learning-based model that explores phenotypic differences while correcting undesired batch effects. The inputs of scFLASH include gene expression profiles, batch IDs, and condition labels of interest (e.g. disease or healthy) for each cell (Fig. 1(a)). scFLASH comprises three main components: a CVAE, an adversarial batch classifier, and a penalized condition classifier (Fig. 1(b)). Using the CVAE framework, scFLASH disentangles the high-dimensional expression profiles with one-hot-encoded vectors of batch IDs into latent known conditional attributes and unknown biological attributes. Through adversarial training with the encoder, the batch classifier corrects the underlying batch effects in these attributes. Importantly, scFLASH introduces a condition classifier with a novel penalty function specifically designed to preserve biological variations flexibly. Benefiting from the above components, scFLASH can facilitate downstream analyses by enhancing gene expression profiles and identifying condition-related cell subpopulations (Fig. 1(c)).

2.2 scFLASH has superior performance in integrating Alzheimer's disease datasets

To evaluate the integration performance of scFLASH, we used a series of single-nucleus RNA sequencing (snRNA-seq) data from Alzheimer's disease (AD) patients under two conditions: AD and control [13]. The initial analysis revealed prominent batch effects in the unintegrated data, which compromised the accurate representation of cellular heterogeneity (Supplementary Fig. 1). We compared scFLASH with approaches specially designed for MBMC integration tasks as well as some commonly used methods for scRNA-seq data integration, such as Seurat [50] and Harmony [19]. A total of six metrics, categorized into three criteria (biological conservation, batch correction, and condition conservation), were employed to evaluate the integration performances from different perspectives (methods).

By leveraging adversarial training and penalty terms, scFLASH achieved the highest overall aggregation score, outperforming the second-best method, scDisInFact, by 19.3%

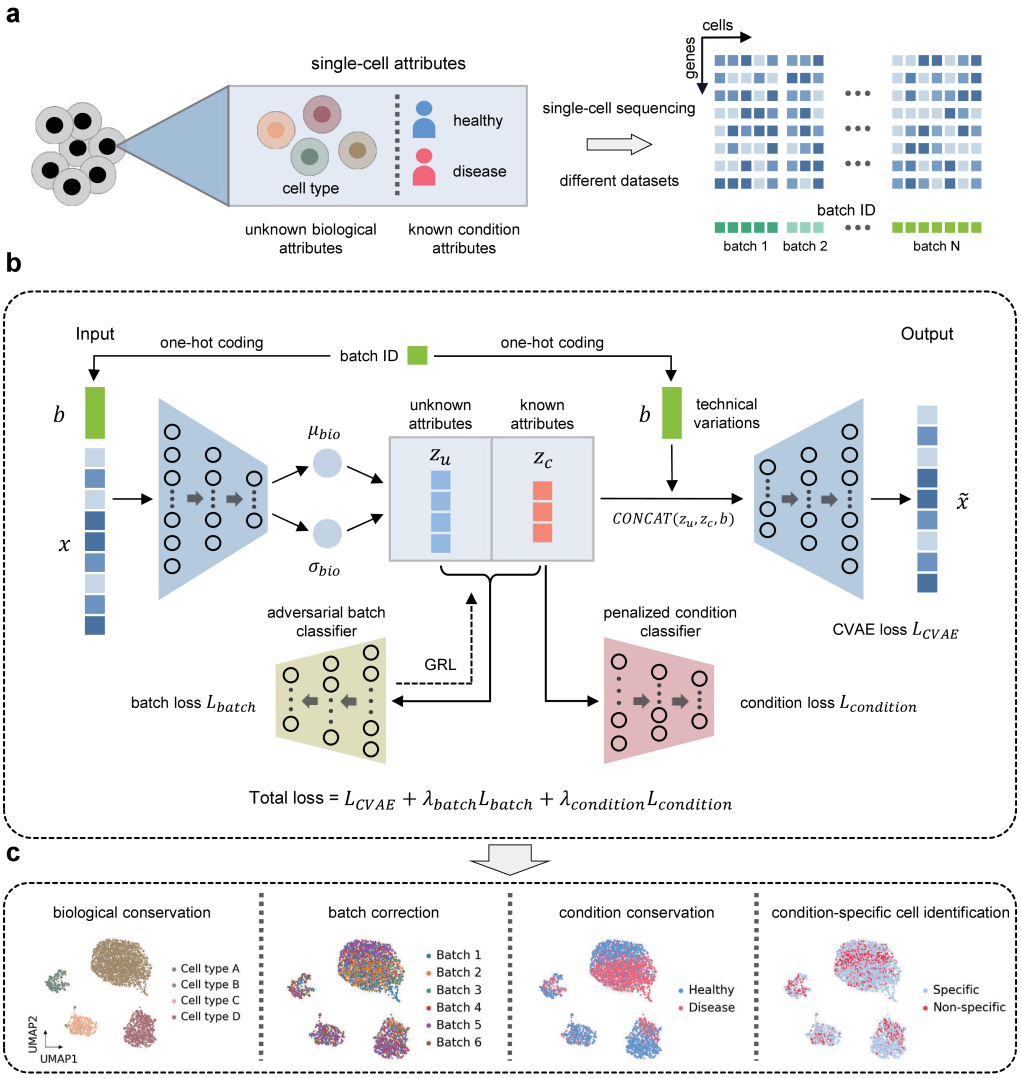

Figure 1: The workflow of scFLASH. (a) The inputs of scFLASH include gene expression profiles, batch IDs, and known condition labels for the cells. (b) scFLASH is a deep learning model with three components: a conditional variational autoencoder, an adversarial batch classifier, and a penalized condition classifier. (c) scFLASH achieves an integration for preserving biological variations across multiple conditions and removing technical noise, facilitating the candidate downstream analyses.

(Fig. 2(a) and Supplementary Table 1). While scDisInFact excelled in clustering, it underperformed in condition conservation, as indicated by the lower cond_knn score, likely due to the limited extraction of condition information. In contrast, scMerge2 and scDisco focused more on preserving condition differences but had relatively inferior batch correction performances. As for the methods such as scDREAMER [46], scVI [26], Harmony, BBKNN [36], Seurat, and Scanorama [15] that did not account for conditions, they per-

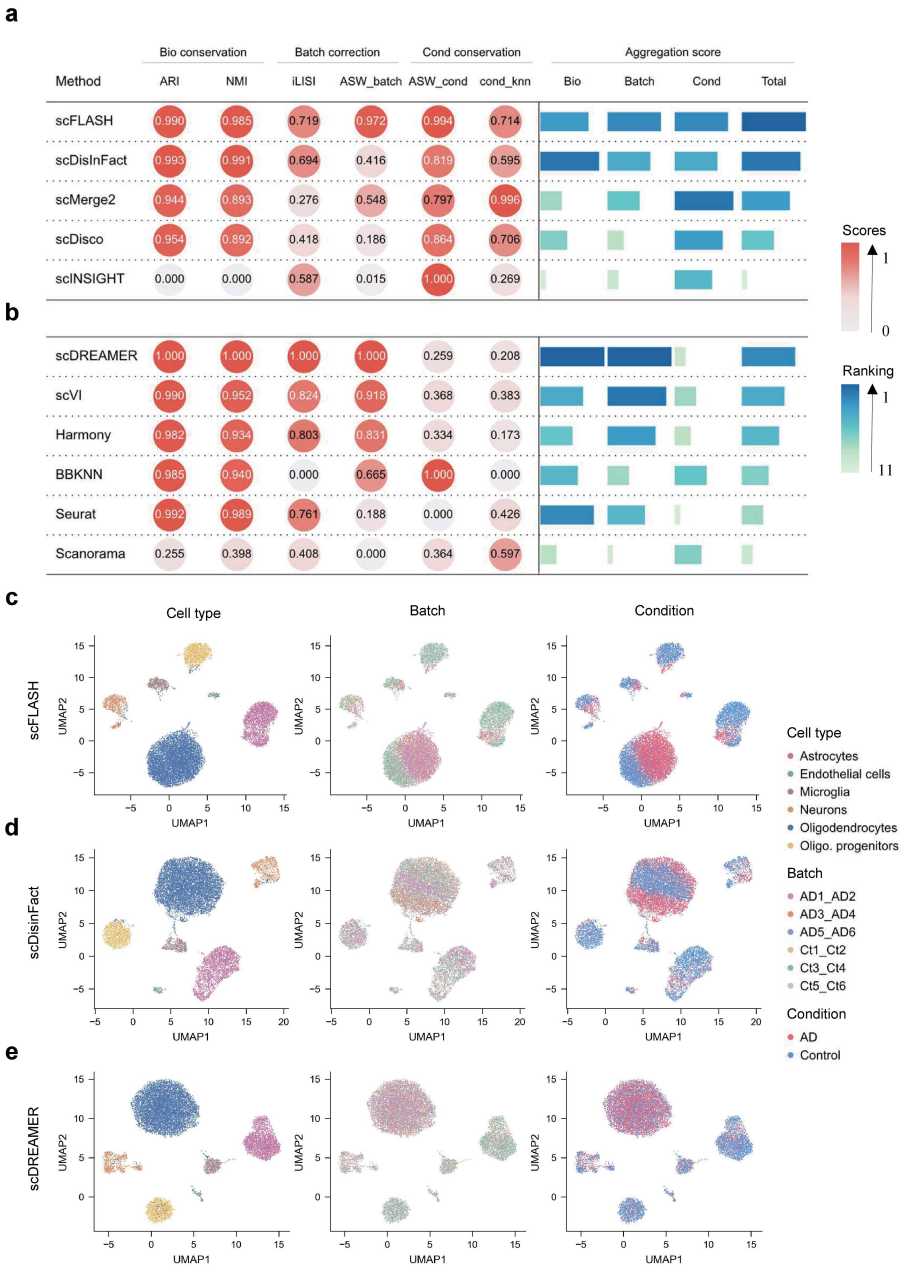

Figure 2: Benchmarking results on Alzheimer's disease dataset. (a)-(b) The quantitative evaluations of biological conservation, batch correction, condition conservation, and aggregation scores in Alzheimer's disease integration task using methods (a) considering conditions and (b) without considering conditions. The scores and metrics across different methods were normalized accordingly with a range from 0 to 1. (c)-(e) UMAP visualizations of Alzheimer's disease dataset using latent space embeddings integrated by (c) scFLASH, (d) scDisinFact, and (e) scDREAMER. The cells were annotated and colored by cell types (left), batches (middle), and conditions (right).

formed well in batch correction and biological conservation but conflated the condition and technical differences (Fig. 2b). Overall, scFLASH enabled a more balanced integration for preserving biological variations and removing technical noise, thus providing a more reliable representation of cellular heterogeneity across different conditions.

We then explored how the cells from different conditions are distributed in each cell type using the uniform manifold approximation and projection (UMAP) visualization. After scFLASH's integration, cells from different batches were well-mixed. Notably, we observed that some of the cells from AD and control were overlapped, and the remaining abundance for condition-specific cells was prominent for almost all cell types (Fig. 2c). For example, the oligodendrocytes exhibited an apparent condition difference and a more precise "boundary" between the condition-specific cells. As for the other top-ranking methods, although scDisInFact successfully distinguished cell types, it embedded control cells within the AD cells and thus may affect the identification of the condition-specific cells (Fig. 2d). scMerge2 overemphasized the condition differences, and the subclusters of AD and control cells had few overlaps (Supplementary Fig. 2). In contrast with scMerge2, scDREAMER integrated cells from various batches well, but it mistakenly merged cells from different conditions into the same cluster (Fig. 2e). In summary, these pieces of evidence showed that scFLASH can preserve the condition differences, making it applicable to identifying condition-specific cells in downstream analyses.

2.3 scFLASH enhances Alzheimer's disease relevant gene detection

By simultaneously correcting batch effects and identifying condition-specific cells, scFLASH enables more precise comparisons of cellular states across conditions. We hypothesized that directly analyzing these condition-specific cells would better characterize the cellular and molecular mechanisms underlying Alzheimer's disease pathogenesis.

We first visualized the locations of the condition-specific cells and observed that the condition-specific cells were mainly distributed away from the boundary between AD and control cells, particularly in oligodendrocytes (Fig. 3(a)). Differential expression analysis showed that directly comparing condition-specific cells can identify more AD upregulated differential expression genes (DEGs) (Fig. 3(b)). We thereby used an external AD bulk dataset [23] to validate whether these DEGs are biologically meaningful (methods). We found that the signature scores of scFLASH-identified DEGs had more statistical significance between AD patients and controls (Fig. 3(c) and Supplementary Fig. 4). Moreover, the signature scores of the scFLASH uniquely identified DEGs could still significantly distinguish AD patients and controls, whereas their counterparts failed to achieve this (Student's t-test $p = 3.81\text{e-}05$ versus 0.369, Fig. 3(c)). We also performed the same analysis pipeline within each cell type. Similarly, scFLASH identified more AD-upregulated DEGs in each cell type (Supplementary Fig. 5 and Supplementary Table 2). Interestingly, the condition differences varied between cell types, with many AD-upregulated genes unique to specific cell types (Fig. 3(d)). Besides, functional enrichment

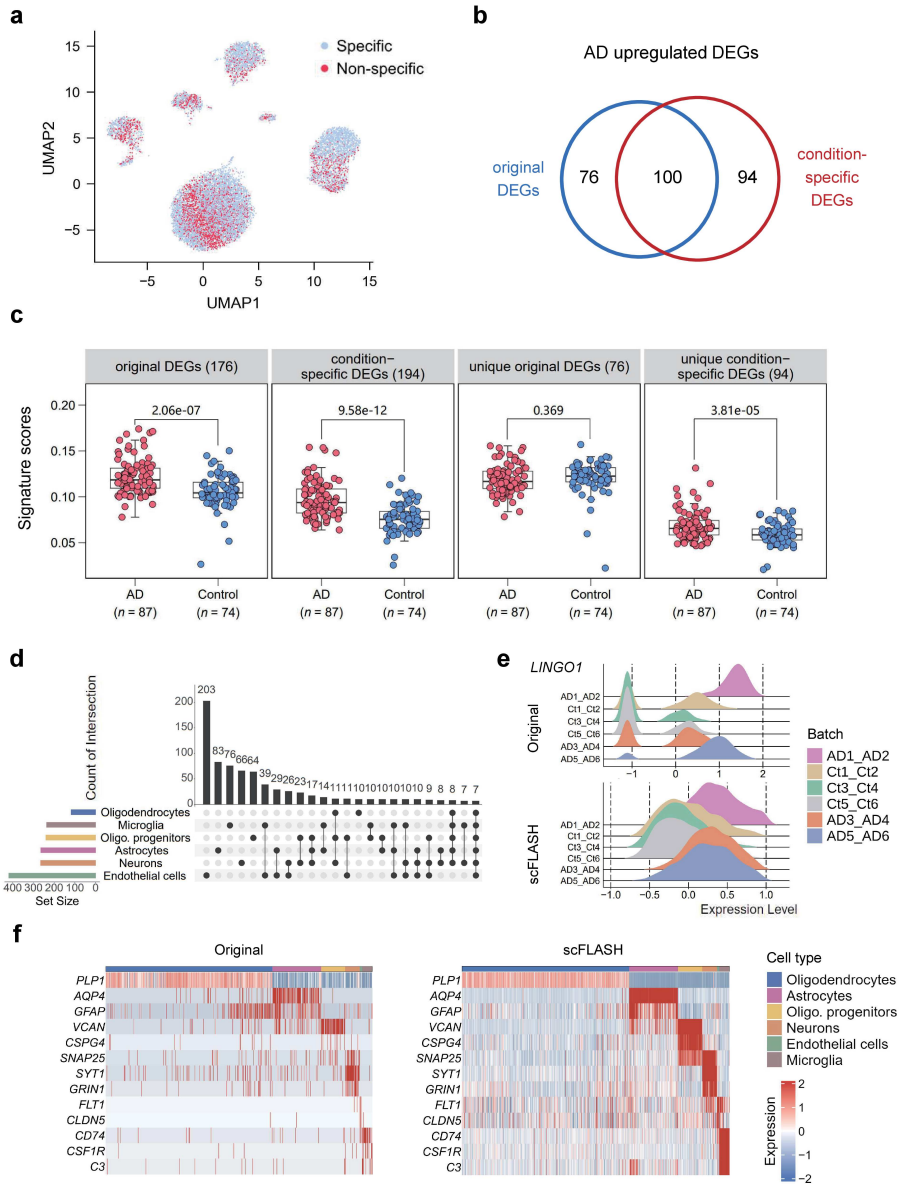

Figure 3: Analyses results of scFLASH on Alzheimer's disease dataset. (a) UMAP visualization of Alzheimer's disease dataset using latent space embeddings integrated by scFLASH. The gray and red dots represent condition-specific cells and others, respectively. (b) Venn diagram shows overlap between upregulated DEGs identified using the original cells and the condition-specific cells. (c) Boxplots show the signature scores of the corresponding DEGs in AD ($n=87$) and control ($n=74$) samples. The box plot center line and the box limits represent median value and upper and lower quartiles, respectively. Box whiskers indicate the largest and smallest values no more than 1.5 times the interquartile range from the limits. The statistical P values were determined by the Student's t-test. (d) Upset plot shows the overlaps of upregulated DEGs in comparing condition-specific cells across different cell types. (e) Ridgeline plots show the expressions of *LINGO1* across different batches in original data (top) and corrected by scFLASH (bottom). (f) Heat maps of known marker genes using the original expressions (left) and the corrected expressions by scFLASH (right).

analysis consistently showed that protein misfolding-related pathways were upregulated in AD cells (Supplementary Fig. 6), aligning with the known pathology of AD, where amyloid plaques formed by misfolded proteins drive disease progression [14,38].

We further explored the performance of scFLASH correction on gene expression profiles. As an example, we selected *LINGO1*, an upregulated gene across multiple cell types (Supplementary Fig. 7) that plays a key role in negatively regulating neuronal survival and axon integrity [1,3,31]. The original expression profiles of *LINGO1* showed evident batch effects. After executing scFLASH, *LINGO1*'s expressions were consistent across different batches, and the distribution differences for each condition became discriminated (Fig. 3(e)). These observations cannot be obtained using other computational approaches like Seurat and scMerge2 (Supplementary Fig. 8). We then explored the expression patterns of known cell type marker genes. Compared to the original data, the expression profiles corrected by scFLASH improved cell type annotations and reduced noise (Fig. 3(f) and Supplementary Fig. 9). For instance, scFLASH revealed a more apparent pattern of the marker gene *FLT1* in the rare endothelial cells, which accounted for only 0.8% of the AD dataset, while other methods detected only weak signals. Thus, we demonstrated that scFLASH effectively identifies condition-specific cells, revealing molecular signatures and differential expression patterns that are more biologically meaningful.

2.4 scFLASH effectively integrates COVID-19 datasets across multiple conditions

scRNA-seq datasets with multiple conditions can exaggerate or obscure the real condition-specific information, making the MBMC integration task more challenging. To explore scFLASH's ability to integrate datasets with multiple conditions, we used a COVID-19 scRNA-seq dataset from peripheral blood mononuclear cells (PBMC) [22]. A total of 58,199 cells were collected from healthy donors and patients with severe, mild, and asymptomatic COVID-19, as well as severe influenza. The original gene expression data showed some clusters driven by cell type, whereas the technical variations between batches still exist (Supplementary Fig. 10). For example, classical monocytes from F1 and F3 in the influenza condition cluster separately, likely due to batch effects rather than biological differences.

Next, we quantitatively evaluated the integration performance of scFLASH with other methods. We found that scFLASH had the best integration performance in this challenging task, achieving a good balance between preserving biological differences across multiple conditions and removing technical noise (Fig. 4(a) and Supplementary Table 3). Although scDisco and scDisInFact performed well in batch correction (especially for iLISI), their relatively lower condition conservation scores suggested their overcorrection of batch effects, potentially masking the condition-specific biological contents. Interestingly, we found that several methods considering conditions performed even worse than the traditional methods, such as Harmony and scVI, indicating that complicated conditions could hinder integration if the method fails to incorporate the condition in-

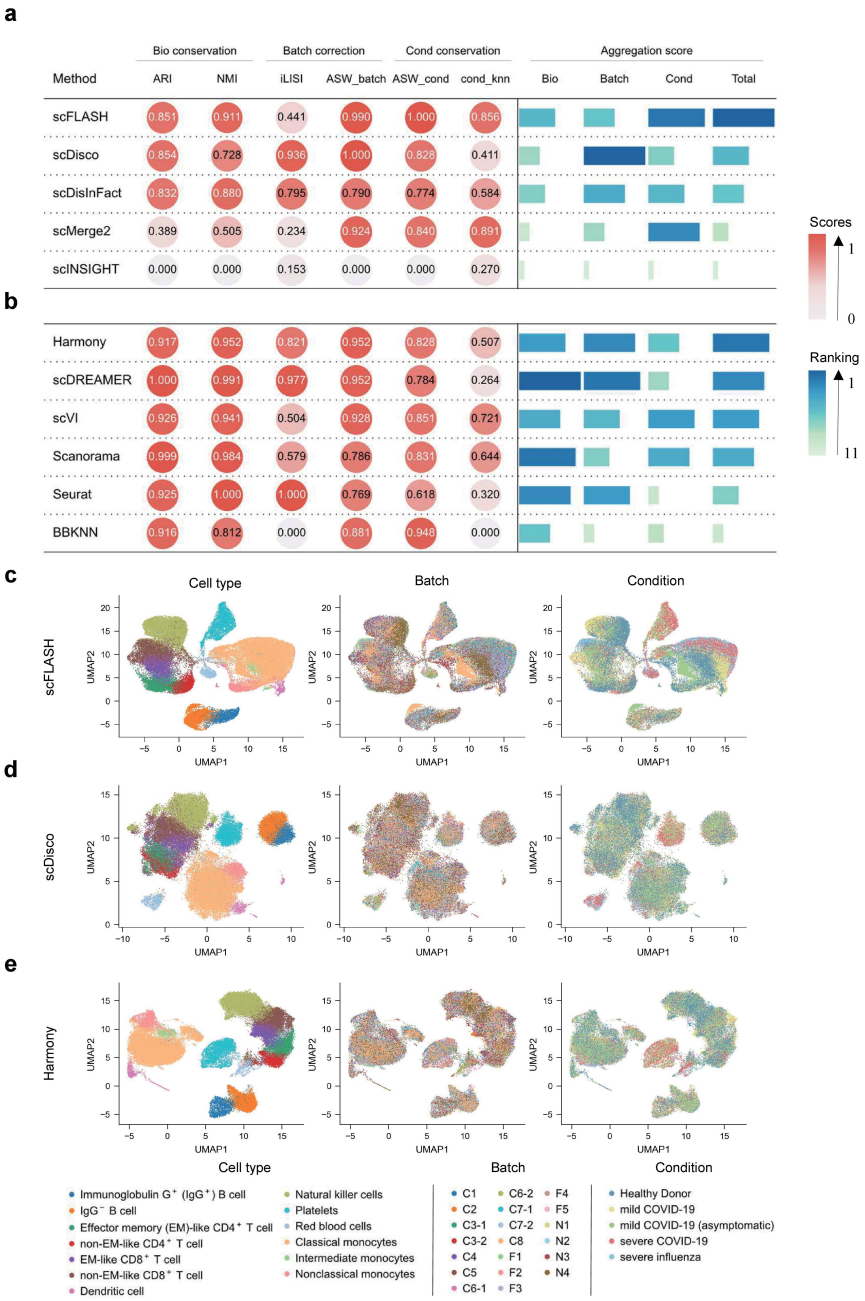

Figure 4: Benchmarking results on COVID-19 dataset. (a)-(b) The quantitative evaluations of biological conservation, batch correction, condition conservation, and aggregation scores in COVID-19 integration task using methods (a) considering conditions and (b) without considering conditions. The scores and metrics across different methods were normalized accordingly with a range from 0 to 1. (c)-(e) UMAP visualizations of COVID-19 dataset using latent space embeddings integrated by (c) scFLASH, (d) scDisco, and (e) Harmony. The cells were annotated and colored by cell types (left), batches (middle), and conditions (right).

formation properly. Among the methods that do not account for conditions, Harmony and scDREAMER ranked first and second in overall aggregation scores. However, these methods cannot preserve necessary condition differences (Fig. 4(b)).

We further explored the cell distributions for all cell types using the UMAP visualization. scFLASH successfully integrated the COVID-19 dataset with the patterns of various cell types to be re-characterized (Fig. 4(c)). Notably, some biological variations from multiple conditions were better distinguished when integrated by scFLASH. For instance, the cells from different conditions in classical monocytes showed a layered structure, indicating that asymptomatic and mild-severe COVID-19 had distinct molecular markers and risk factors when affected by the virus [53]. The above layered structure was not observed when integrated by Harmony and scDisco (Figs. 4(d), 4(e)). As for other methods that consider conditions, scDisInFact identified a similar layered structure in platelets, but it mixed cells from healthy samples with other conditions for the remaining cell types. scMerge2 was unable to integrate batch F1 and F3 with other batches, resulting in incomplete batch correction (Supplementary Fig. 11). In conclusion, scFLASH effectively integrated the complex datasets from multiple conditions by correcting batch effects while preserving condition differences.

2.5 scFLASH characterizes the transcriptional identities of classical monocytes under different COVID-19 conditions

In this section, we analyzed the COVID-19 datasets integrated by scFLASH to characterize the transcriptional identities across different conditions. Noticed that the classical monocytes ranked the top in condition-specific prioritizations predicted by Augur [47] (Supplementary Fig. 13), and the inflammatory cytokines secreted by classical monocytes are thought to play a key role in the progression of COVID-19 [41], we thereby focused our analyses on classical monocytes.

We first identified the upregulated DEGs in classical monocytes between COVID-19 (mild and severe), influenza, and asymptomatic cases with healthy individuals using the expression profiles corrected by scFLASH. We found that 68 genes were upregulated in both COVID-19 and influenza, suggesting shared mechanisms in triggering host immune responses (Fig. 5(a)). Besides, we adopted the cytokine-responsive gene sets from Library of Integrated Network-based Cellular Signatures (LINCS) [9] to perform the functional enrichment analysis for the identified DEGs. We observed that the DEGs related to COVID-19 and influenza were enriched for tumor necrosis factor-α/interleukin-1 (TNF-α/IL-1) responsive genes (Fig. 5(b) and Supplementary Table 4), indicating a stronger inflammatory response in symptomatic progression since TNF-α drives inflammation, which is critical during severe infections [60]. In contrast, asymptomatic cases showed lower levels of enrichment of TNF-α and IL-1. Additionally, compared to influenza, COVID-19 had significant enrichment of type I interferon-γ (IFN-γ) responsive genes, a pathway known to exacerbate inflammation in severe COVID-19 cases [22, 49].

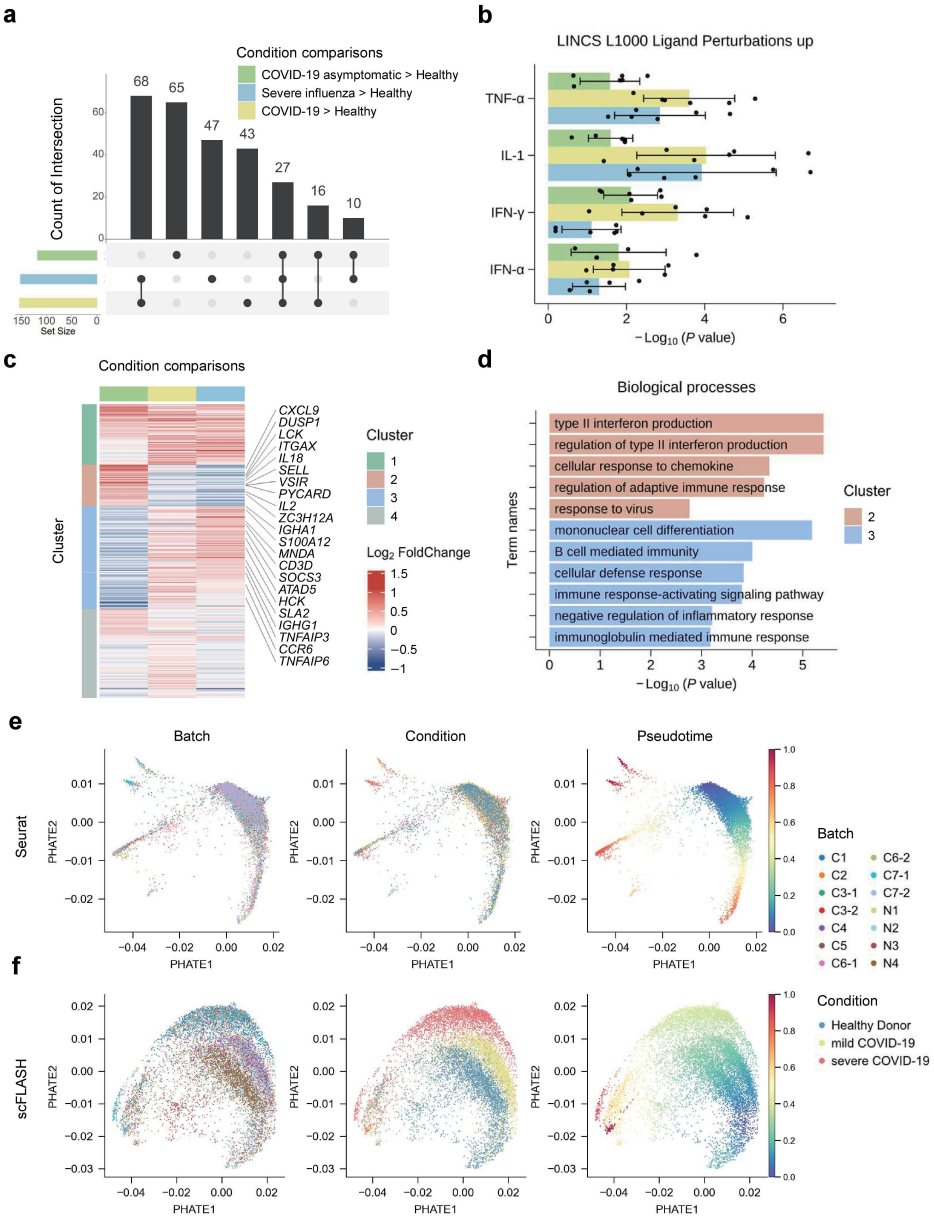

Figure 5: Analyses results of scFLASH on COVID-19 dataset. (a) Upset plot shows the overlaps of upregulated DEGs in classical monocytes between COVID-19 (mild and severe), influenza, and asymptomatic cases with healthy individuals. (b) Enrichment bar plots of cytokine-responsive gene sets from LINCS using DEGs from different comparisons. The bars represent the standard errors. (c) Heat map of differential gene fold changes in disease cases versus healthy individuals across different comparisons. The genes were grouped into four clusters using the K-means clustering. (d) Enrichment bar plots of selected biological processes pathways enriched by cluster 2 and cluster 3 genes. (e)-(f) PHATE visualizations of classical monocytes integrated by (e) Seurat, (f) scFLASH. The cells were annotated and colored by batches (left), conditions (middle), and pseudotime (right).

To better characterize the transcriptional features of classical monocytes, we performed K-means clustering on the above DEGs. These genes were grouped into four clusters, allowing us to explore the specificity and associations between disease states. For instance, cluster 2 is specific to asymptomatic cases, while cluster 3 is associated with both influenza and COVID-19 (Fig. 5(c)). Besides, we observed high expression of the IFN-γ-dependent chemokine *CXCL9* in cluster 2, aligning with previous findings that asymptomatic patients have higher *CXCL9* levels than healthy controls [30]. We further performed functional enrichment analysis and discovered the differences between clusters: cluster 2 genes were associated with interferon and antiviral responses, whereas cluster 3 genes were related to immune cell differentiation and activation (Fig. 5(d) and Supplementary Fig. 14).

Next, to further investigate the patterns of change associated with COVID-19 severity, we performed pseudotime inference and compared our results with Seurat. Specifically, we selected healthy, mild, and severe COVID-19 cells within classical monocytes for pseudotime inference. We found that the cells from different conditions were mixed when using the data integrated by Seurat (Fig. 5(e)). In contrast, scFLASH better aligned classical monocytes along the disease severity spectrum, with severe cases mapped to later stages and healthy cases to earlier stages (Fig. 5(f)). A similar result was also observed using CD4 T cells from the mild and severe COVID-19 cases, revealing a connection of pseudotime with different disease states (Supplementary Fig. 15).

Collectively, scFLASH can reveal meaningful transcriptional identities across different conditions, facilitating downstream analyses such as immune landscape reconstruction and pseudotime inference.

2.6 scFLASH removes batch effects from different donors and preserves conditional information in diabetes

In real applications, the batch effects can be originated from various sources. To demonstrate the versatility of scFLASH in handling different types of batch effects, we collected 222,077 human islet cells from 67 donors within the Human Pancreas Analysis Program (HPAP) [16, 45]. These cells were categorized into four groups: non-diabetic, type 1 diabetes autoantibody-positive (AAb+), type 1 diabetes (T1D), and type 2 diabetes (T2D). UMAP visualizations showed that scFLASH effectively mixed the 67 batches while preserving the heterogeneity of different cell types (Fig. 6(a)). Additionally, compared to the original data, the expression profiles corrected by scFLASH better captured intercellular variations and reduced the noise (Supplementary Fig. 16).

Given that the dysfunction of beta cells is a key pathological feature of both type 1 and type 2 diabetes [10], we analyzed the transcriptional differences in beta cells across different conditions. When comparing the condition-specific cells obtained from T2D and non-diabetic individuals, scFLASH identified 147 upregulated DEGs in T2D, which were significantly enriched in biologically meaningful pathways, such as endocrine system development and pancreatic A cell differentiation (Figs. 6(b), 6(c) and Supplementary Table 5). Notably, these pathways cannot be identified using the standard differentially

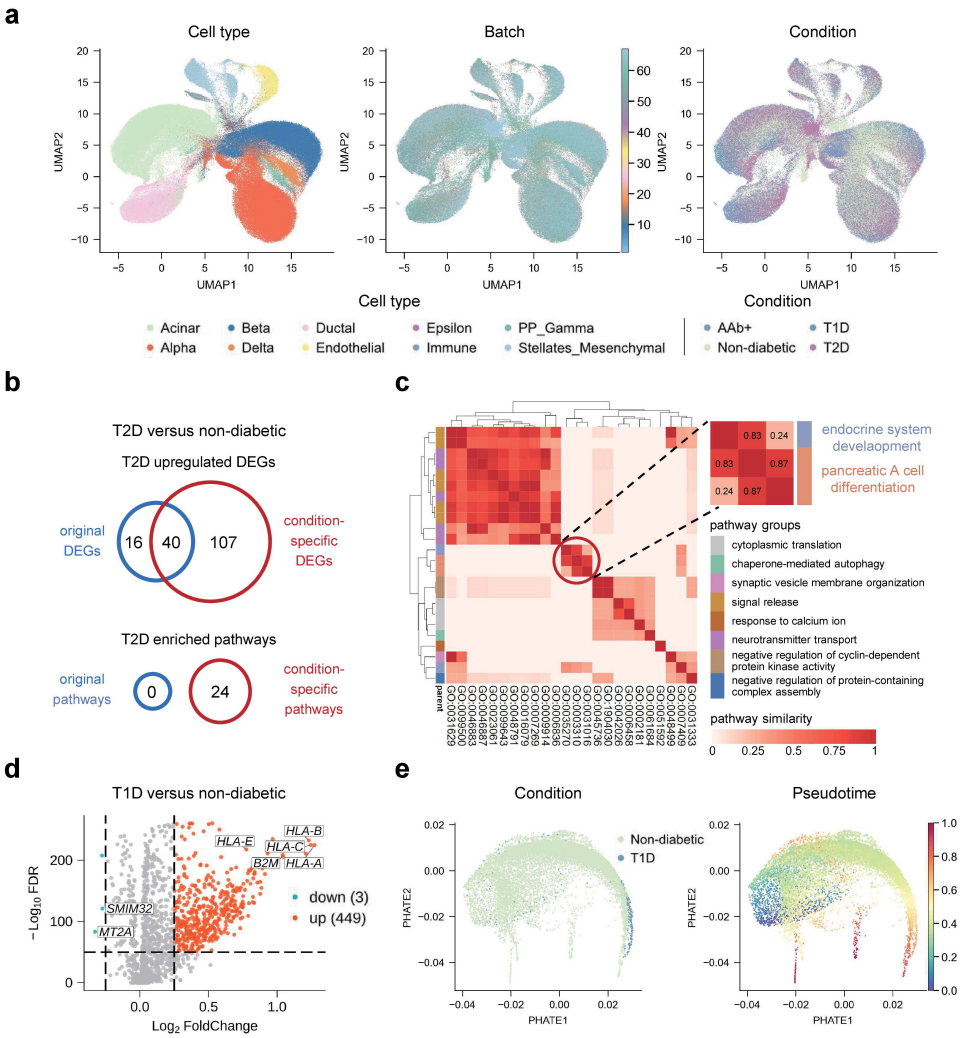

Figure 6: Analyses results of scFLASH on diabetes dataset. (a) UMAP visualizations of diabetes dataset using latent space embeddings integrated by scFLASH. The cells were annotated and colored by cell types (left), batches (middle), and conditions (right). (b) Venn diagram shows overlaps between T2D upregulated DEGs (top) and enriched GO terms (bottom) identified using the original cells and the condition-specific cells. (c) Heat map shows the similarities between the enriched GO terms of upregulated DEGs identified by scFLASH. (d) Volcano plot of DEGs in T1D versus non-diabetic. The two vertical dashed lines represent ± 0.25 log-transformed fold changes in gene expression, and the horizontal dashed line denotes an FDR cutoff of 1e-50. The FDR was the adjusted P value calculated by the Wilcoxon rank-sum test. (e) PHATE visualizations of control and T1D beta cells integrated by scFLASH. The cells were annotated and colored by condition (left) and pseudotime (right).

expressed gene analysis pipelines with the original expression profiles in contrast. As for T1D, we identified a greater number of upregulated genes when compared with T2D, including major histocompatibility complex (MHC) class I genes (such as *HLA-A* and

HLA-B) and MHC-related genes such as *B2M* [10, 33] (Fig. 6(d), Supplementary Fig. 17, and Supplementary Fig. 18). Interestingly, MHC class I gene expression showed minimal changes in AAb+ patients, suggesting that beta cells in these individuals are less affected, which aligns with a recent study [10].

Next, we conducted a pseudotime analysis to explore the capacity of scFLASH to handle phenotypically unbalanced data. In all 44,559 beta cells, T1D cells and T2D cells account for 1.6% and 18.3%, respectively (Supplementary Fig. 19(a)). Although the disease cells comprised relatively lower percentages, scFLASH successfully identified condition-specific differences during time varies. Specifically, we found that scFLASH mapped disease cells to later stages, effectively capturing trajectories of disease development (Fig. 6(e) and Supplementary Fig. 19(b)). These observations suggest that scFLASH gives potential biological insights into diabetes and can help to provide a deeper understanding of the underlying mechanisms of the disease progression.

2.7 scFLASH removes batch effects from different studies and is applicable to large-scale integration tasks

With the accumulation of single-cell data, there is an urgent need for computational approaches that can be efficiently applied to large-scale data analysis. To demonstrate the scalability of scFLASH, we integrated a COVID-19 PBMC atlas containing approximately 6.7 million cells from 2,162 samples across 25 different studies [8]. UMAP visualizations showed that batch effects from different studies fragmented some cell types into multiple clusters (Fig. 7(a)). After performing scFLASH, our method successfully grouped the major cell types, such as T cells and B cells, while preserving the condition differences (Fig. 7(b) and Supplementary Fig. 20). Importantly, although we used different studies as batch labels, scFLASH can also eliminate batch effects caused by different sequencing technologies and sample variations (Supplementary Fig. 21). Besides, scFLASH is applicable for integrating different conditions within the same batch. For example, UMAP visualizations stratified by the studies showed that scFLASH effectively distinguished phenotypic differences in studies such as Zhang2021, Tsang2021, and Meyer2021 (Supplementary Fig. 22).

The pathogenesis of COVID-19 is closely associated with abnormal innate immune responses within cells, such as type I interferons (IFN) [35, 41, 49]. To investigate whether scFLASH-derived cell types have specific immune responses in COVID-19, we generated pseudo-bulk profiles for each sample under given cell types and calculated the corresponding interferon response scores (methods). Our analysis showed that under COVID-19, monocytes and dendritic cells exhibited the most elevated scores, reflecting their known role in interferon-mediated viral immune responses [49] (Fig. 7(c)). Functional enrichment analysis of the disease-upregulated genes revealed that these genes are involved in inflammation-related pathways such as viral response and regulation of the viral process, consistent with a previous research [35] (Fig. 7(d), Supplementary Fig. 23, and Supplementary Table 6). In addition, genes associated with Type I/III interferon re-

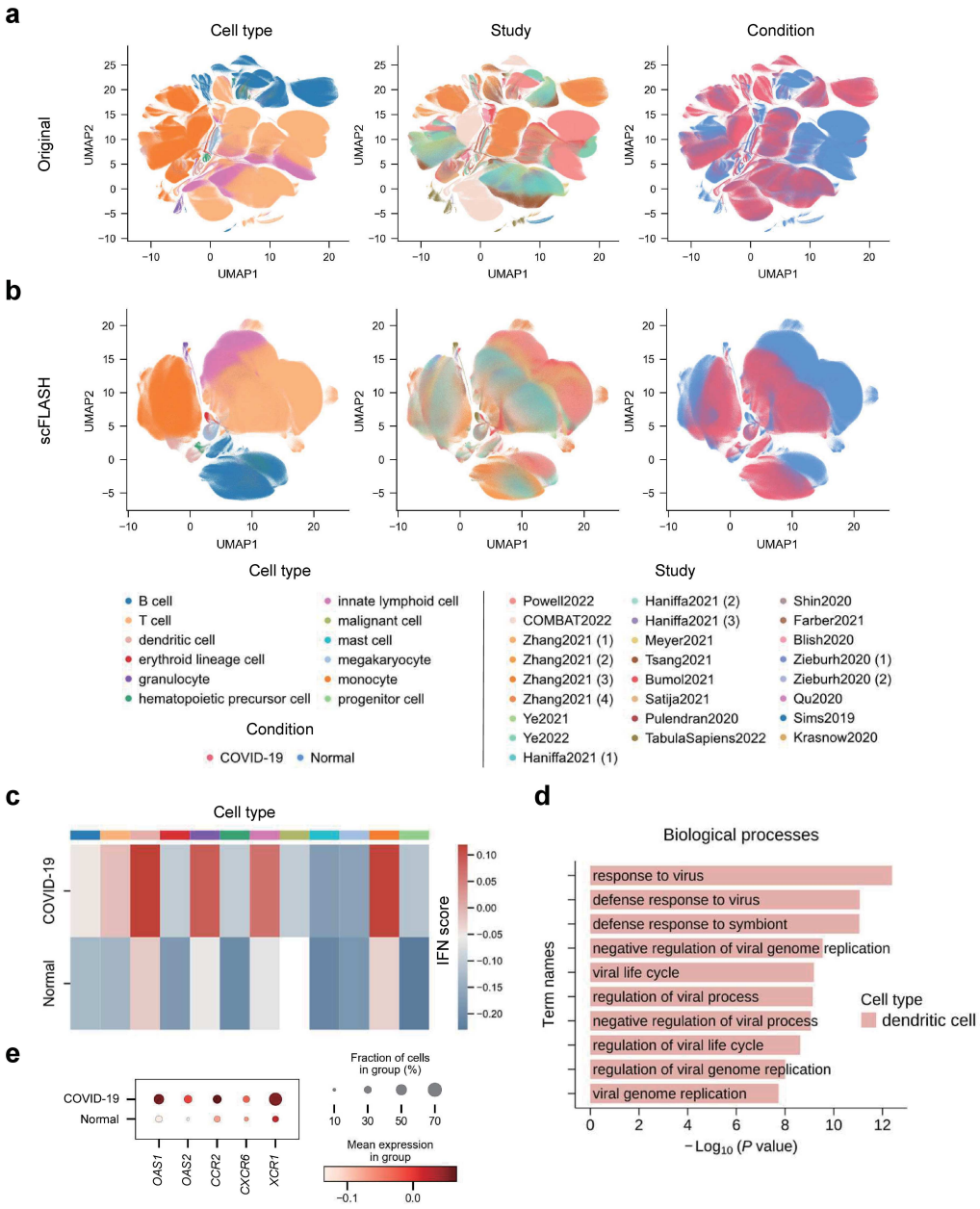

Figure 7: Analyses results of scFLASH on PBMC atlas. (a)-(b) UMAP visualizations of (a) the unintegrated PBMC atlas and (b) the latent space embeddings integrated by scFLASH. The cells were annotated and colored by cell types (left), batches (middle), and conditions (right). (c) Heat map of interferon response scores across different cell types and conditions. (d) Enrichment bar plots of selected biological processes pathways using COVID-19 upregulated DEGs in dendritic cells. (e) Dot plot of the expressions of the selected genes. The dot color is scaled by mean expression, and the dot size is proportional to the fraction of cells with corresponding gene expressed.

sponses were highly expressed in COVID-19 cells, a similar observation with previous studies on COVID-19 susceptibility [34, 43] (Fig. 7(e)). Besides, scFLASH had comparable runtimes with other integration methods with cell numbers ranging from 10,000 to 100,000 (Supplementary Fig. 24).

Overall, scFLASH showed strong potential as a scalable tool for large-scale single-cell analysis, providing efficient batch correction for diverse batch sources.

3 Discussion

In this study, we presented an advanced deep learning-based model, scFLASH, to explore phenotypic differences while correcting undesired batch effects. scFLASH integrates multi-batch and multi-condition single-cell datasets using a conditional variational autoencoder framework, effectively decoupling high-dimensional data into latent known conditional attributes and unknown biological attributes. An adversarial batch classifier is employed to correct batch effects, and a penalized condition classifier is introduced to preserve biological variations across conditions. Our results showed that scFLASH had superior integration performances and outperformed existing methods in biological conservation, batch correction, and condition conservation.

Through comprehensive analyses of related datasets, we highlighted the advantages of scFLASH. First, scFLASH is a versatile model designed explicitly for MBMC integration tasks. It seamlessly integrates datasets with two or more conditions while correcting batch effects from diverse sources, such as donors or samples, sequencing technologies, and studies. Additionally, scFLASH's scalability enables it to handle large-scale scRNA-seq datasets efficiently, ensuring accurate and consistent representations of cellular and molecular features across multiple conditions and diverse experimental settings.

A notable innovation of scFLASH is identifying condition-specific cell subpopulations. In case-control scRNA-seq analyses, sample-level labels are often directly transferred to individual cells, assuming all cells from the case sample are equally affected. However, the condition may impact only a small subset of cells, making the above procedure less effective in discovering biologically meaningful subpopulations [12]. Besides, scFLASH does not follow the way of differential abundance analysis algorithms such as DAseq [61], Milo [7], and MELD [5], which do not consider the technical variability across samples and require separate integration steps to identify phenotype-associated subpopulations. In contrast, scFLASH integrates batch correction and subpopulation identification within a unified framework, ensuring excellent reliability and efficiency in real applications.

Additionally, scFLASH excels in denoising and uncovering meaningful transcriptional features across different conditions. By refining original expression profiles at the gene level, scFLASH reveals biologically significant transcriptional identities, enabling detailed characterizations of cellular states. This capability facilitates a wide range of downstream analyses, such as immune landscape reconstruction, pseudotime trajectory

inference, and the identification of DEGs, providing deeper insights into cellular and molecular mechanisms under diverse conditions.

There may still be room to improve the MBMC integration performance of scFLASH. For example, the success of scFLASH is mainly attributed to the usage of the condition classifier, which introduces a novel penalty function specifically designed to preserve the biological variations flexibly. The current scFLASH uses a condition constraint factor k to control the detection of phenotypic differences. Designing an adaptive way to select k automatically may further improve the integration performance. Additionally, our ablation studies showed that scFLASH sufficiently utilized both condition and batch information (Supplementary Table 10, Supplementary Fig. 3, and Supplementary Fig. 12). In further, we will explore whether adding prior information, such as cell type, tissue origin, and differentiation stage, would enhance the characterization of cellular states.

Moreover, while our current framework focuses on integrating scRNA-seq data, scFLASH may also be applied to other integration tasks considering condition and batch information. For instance, we can decipher phenotype-associated cell subpopulations using single-cell transcriptome with chromatin accessibility data from different conditions [51] or apply scFLASH to spatially resolved transcriptomics with multiple sections to detect anomalous tissue domains [58].

Overall, scFLASH demonstrates promise for integrating multi-batch and multi-condition single-cell datasets with advanced analysis to dissect disease-relevant genes and identify phenotype-associated cell subpopulations. We hope that scFLASH is a powerful addition to the current scRNA-seq analysis arsenal and can offer biological insights into the cellular mechanisms of disease pathology through broad applications.

4 Methods

4.1 Data pre-processing

In this study, we followed the standard pipeline of data preprocessing provided by Scanpy [54]. First, we filtered out genes expressed in fewer than three cells and excluded cells with fewer than 200 expressed genes. Next, the expression matrix was normalized and log-transformed. Finally, the top highly variable genes were selected and scaled to unit variances and zero means. The detailed information of datasets used in our study was summarized in Supplementary Table 7.

4.2 The scFLASH model

scFLASH is a deep learning-based model comprising three main components: a conditional variational autoencoder (CVAE), an adversarial batch classifier, and a penalized condition classifier. The CVAE extracts latent biological attributes, separating them into known conditional and unknown biological attributes. The batch classifier removes batch effects from all latent attributes, while the penalized condition classifier ensures

that the condition attributes retain information about known conditions. With the combined functionality of these three components, scFLASH effectively removes batch effects from the data while preserving its conditional information. In the following sections, we introduce each component's mechanisms and details of implementation.

4.3 Conditional variational autoencoder

CVAE extends the variational autoencoder [18], incorporating additional conditional variables to guide the generative process. Based on CVAE, scFLASH explicitly models the relationships between the input expression profiles and the batch IDs, ensuring that the latent space embeddings contain biological signals while isolating technical variations.

We denote the preprocessed single-cell gene expression profile as x and one-hot-encoded batch ID as b, respectively. scFLASH concatenates x and b and passes them through the encoder network E. scFLASH learns latent space embeddings defined by the mean μ_{bio} and variance σ^2_{bio}, parameterized as $\mu_{bio},\sigma^2_{bio}=E(x,b;\theta)$, where θ is the parameters of E. The latent biological attributes z_{bio} are re-sampled as

$$z_{bio}=\sigma^2_{bio}\epsilon_{bio}+\mu_{bio},\quad \epsilon_{bio}\sim\mathcal{N}(0,I). \tag{4.1}$$

We divide z_{bio} into two parts: the first three-quarters represent unknown biological attributes z_u, and the remaining one-quarter represent condition attributes z_c. This relationship can be formulated as $(z_u,z_c)=z_{bio}$.

The input of the decoder network D is z_{bio} and b, producing the reconstructed expression profile $\tilde{x}$ as $\tilde{x}=D(z_{bio},b;\phi)$, where ϕ denotes the parameters of D. To optimize the encoder E and decoder D, scFLASH employs a variational inference approach by maximizing the evidence lower bound loss (ELBO) function

$$ELBO(\theta,\phi)=\mathbb{E}_{q_\theta(z_{bio}|x,b)}[\log p_\phi(x\,|\,z_{bio},b)]-\lambda_z KL\big(q_\theta(z_{bio}\,|\,x,b)\,\|\,p(z_{bio})\big). \tag{4.2}$$

The first term encourages the model to reconstruct the input x accurately based on z_{bio} and b. The second term is implemented as the Kullback-Leibler (KL) divergence, quantifying the difference between the posterior distribution $q_\theta(z_{bio}\,|\,x,b)$ and the prior $p(z_{bio})=\mathcal{N}(0,I)$. λ_z is a hyper-parameter balancing these two terms. Using the ELBO function, the CVAE loss can be expressed as

$$L_{CVAE}(\theta,\phi)=-\mathbb{E}_{q_\theta(z_{bio}|x,b)}[\log p_\phi(x\,|\,z_{bio},b)]+\lambda_z KL\big(q_\theta(z_{bio}\,|\,x,b)\,\|\mathcal{N}(0,I)\big). \tag{4.3}$$

In practice, the reconstruction loss (the first term of L_{CVAE}) is calculated using the mean squared error, which evaluates the similarities between the reconstructed output $\tilde{x}$ and the original input x.

4.4 Adversarial batch classifier

An adversarial batch classifier is employed to eliminate batch effects from the latent biological attributes z_{bio}. We put z_{bio} through a gradient reversal layer (GRL) [11] and then feed it into the batch classifier B, which outputs probabilities to predict the batch of each cell

$$P^{(b)} = B\big(GRL(z_{bio});\psi\big), \tag{4.4}$$

where ψ is the parameters of B and $P^{(b)}$ is a matrix with dimensions $n \times n_b$, where n and n_b denote the number of cells and batches, respectively. Each element $P_{ij}^{(b)}$ in $P^{(b)}$ is the probability that cell i is predicted to be from batch j. The GRL function has the same input and output during forward propagation but reverses the gradient during back-propagation

$$GRL(z_{bio}) = z_{bio}, \quad \frac{\partial GRL(z_{bio})}{\partial z_{bio}} = -1. \tag{4.5}$$

We use a weighted cross-entropy loss to address the varying amounts of data across different batches

$$L_{batch}(\theta,\psi) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n_b} w_j^{(b)} y_{ij}^{(b)} \log\big(P_{ij}^{(b)}\big), \tag{4.6}$$

where $y_{ij}^{(b)} \in \{0,1\}$ represents the true batch label for each cell. Specifically, $y_{ij}^{(b)} = 1$ means cell i is from batch j, and $y_{ij}^{(b)} = 0$ otherwise. $w_j^{(b)}$ is the reciprocal of the proportion of batch j in all integrated datasets, making L_{batch} assign larger weights to batches with relatively smaller proportions to address the imbalance during training. During the global optimization, the encoder E serves as a generator to maximize L_{batch}, while the batch classifier B serves as a discriminator to minimize L_{batch}, enabling an adversarial training of scFLASH to eliminate batch effects.

4.5 Penalized condition classifier

We adopt a penalized condition classifier to preserve condition-specific characteristics in z_c. We put z_c in the condition classifier C, which outputs probabilities to predict the condition categories of each cell

$$P^{(c)} = C(z_c;\omega), \tag{4.7}$$

where ω is the parameters of C and $P^{(c)}$ is a matrix with dimensions $n \times n_c$, where n_c is the number of condition categories. Each element $P_{ij}^{(c)}$ in $P^{(c)}$ is the probability that cell i is predicted to have the condition j. The condition loss function $L_{condition}$ contains an entropy loss L_e and a penalty loss L_p, balanced by hyper-parameters λ_e and λ_p

$$L_{condition}(\theta,\omega) = \lambda_e L_e(\theta,\omega) + \lambda_p L_p(\theta,\omega). \tag{4.8}$$

Denote $L_{ce}(i)$ as the normalized cross-entropy loss for cell i between $P_{ij}^{(c)}$ and the true condition label

$$L_{ce}(i)=\frac{-\sum_{j=1}^{n_c} y_{ij}^{(c)}\log\left(P_{ij}^{(c)}\right)}{\varepsilon-\sum_{i=1}^{n}\sum_{j=1}^{n_c} y_{ij}^{(c)}\log\left(P_{ij}^{(c)}\right)}, \tag{4.9}$$

where $y_{ij}^{(c)}\in\{0,1\}$ represents the true condition label for each cell. Specifically, $y_{ij}^{(c)}=1$ means cell i has condition k, and $y_{ij}^{(c)}=0$ otherwise. $\epsilon=10^{-6}$ is added to prevent numerical instability caused by extreme values. We calculate the entropy loss L_e as

$$L_e(\theta,\omega)=-\frac{\sum_{i=1}^{n} L_{ce}(i)\log\left(L_{ce}(i)\right)}{\log(n)}. \tag{4.10}$$

A lower entropy indicates a more discriminative condition prediction. The purpose of L_e is to ensure that z_c can not only capture the condition information but also facilitate the selection of cells with condition-specific characteristics (Supplementary Table 10).

We use R_c to quantify how well the condition information is predicted. R_c is calculated as

$$R_c=\frac{1}{1-1/n_c}\cdot\frac{1}{\sum_{i=1}^{n} w_i^{(c)}}\cdot\sum_{i=1}^{n} w_i^{(c)}(1-t_i), \tag{4.11}$$

where $t_i=\sum_{j=1}^{n_c} y_{ij}^{(c)}P_{ij}^{(c)}$, and $1-t_i$ reflects the prediction error probability for cell i. $w_i^{(c)}$ is the reciprocal of the proportion of cells with the same condition as cell i, balancing the contributions of different conditions. $1-1/n_c$ represents a specific scenario in which the prediction error probability for all cells is the same (i.e. uniform distribution). Thus, R_c calculates the ratio between the weighted average prediction error and the prediction error under the specific scenario.

We introduce a condition constraint factor $k\in(0,+\infty]$ to control R_c. The penalty loss L_p is defined as

$$L_p(\theta,\omega)=\max(0,R_c-k). \tag{4.12}$$

A smaller k can obtain a higher prediction accuracy from C, forcing more cells to be predicted to their true condition label. Conversely, a larger k relaxes this requirement. When k is sufficiently large, L_p becomes zero, turning off the condition classifier and making scFLASH focus on batch correction. The default value for k is set to 0.5.

4.6 Implementation of scFLASH

The total loss of scFLASH is the weighted sum of L_{CVAE}, L_{batch}, and $L_{condition}$. All parameters in the neural network (including encoder E, decoder D, batch classifier B, and condition classifier C) are optimized under

$$\min_{\theta,\phi,\omega}\max_{\psi} L_{CVAE}(\theta,\phi)-\lambda_{batch}L_{batch}(\theta,\psi)+\lambda_{condition}L_{condition}(\theta,\omega), \tag{4.13}$$

where λ_{batch} and $\lambda_{condition}$ are hyper-parameters. The detailed information of neural network architecture was summarized in Supplementary Table 8.

In this study, we used the Adam optimizer [17] with the learning rate initially set to 0.001 and increased with iterations. For the large-scale PBMC atlas dataset, the initial learning rate was set to 0.0001 to ensure stable training. The batch size and the total number of epochs were set to 300 and 150, respectively. As for the hyper-parameters, we set

$$\lambda_z = 10^{-4}, \quad \lambda_e = 0.01, \quad \lambda_p = 10, \quad \lambda_{batch} = \frac{2}{1+e^{-10\cdot ep/nep}} - 1, \quad \lambda_{condition} = 1,$$

where ep and nep denote the current epoch and the total number of epochs [11], respectively.

4.7 Construction of the corrected expression profiles and clustering

In order to obtain the corrected expression profile for each cell, after training, we first concatenated the learned biological attributes z_{bio} and the one-hot-encoded batch ID b^* of the batch containing the most cells, allowing cells from different batches to be aligned with the same batch domain. Next, we put z_{bio} and b^* to the decoder D with the learned parameters $\hat{\phi}$

$$\hat{x} = D\left(z_{bio}, b^*; \hat{\phi}\right). \tag{4.14}$$

The reconstructed outputs $\hat{x}$ were used as the scFLASH-corrected expression profiles.

Using the corrected expression profiles, we performed the Louvain clustering at resolutions ranging from 0.1 to 2 in increments of 0.1, and the result with the highest normalized mutual information (NMI) was reported.

4.8 Identification of the condition-specific cell subpopulations.

After training, we used the penalized condition classifier C to identify the condition-specific cell subpopulations. Specifically, we first used the learned condition attributes z_c as the input of C with the learned parameters $\hat{\omega}$

$$\hat{P}^{(c)} = C(z_c; \hat{\omega}). \tag{4.15}$$

Using the same notations, we calculated $\hat{t}_i = \sum_{j=1}^{n_c} y_{ij}^{(c)} \hat{P}_{ij}^{(c)}$, where $\hat{P}_{ij}^{(c)}$ is the element in $\hat{P}^{(c)}$. Similarly, $1-\hat{t}_i$ reflects the prediction error probability for cell i.

We defined the condition-specific confidence (CSC) as

$$CSC_i = \frac{1-\hat{t}_i}{1-1/n_c} - k. \tag{4.16}$$

This score quantifies the ability of a cell that can be correctly assigned to its true condition label. Therefore, the cells with CSC scores less than zero were identified as condition-specific cell subpopulations and marked as non-specific cells otherwise.

4.9 Evaluation metrics

We evaluated the integration performances using three criteria: biological conservation, batch correction, and condition conservation [27]. Specifically, the adjusted Rand index (ARI) [39] and the NMI were used as metrics for biological conservation. The integration local inverse Simpson's index (iLISI) [19] and the average silhouette width across batches (ASW_batch) were employed to evaluate the batch effect removal. As for the condition conservation, the average silhouette width across conditions (ASW_cond) and the F1 score of the k-nearest neighbor classifier (cond_knn) were adopted. Details about these metrics were described in Supplementary Materials.

We computed an aggregation score to represent the overall performance across all three aspects. The scaled value for each metric was obtained through min-max normalization across all competing methods and the final aggregation score was calculated as the arithmetic average of the above six metrics

$$\text{Aggregation Score} = \frac{\text{ARI}+\text{NMI}+\text{iLISI}+\text{ASW\_batch}+\text{ASW\_cond}+\text{cond\_knn}}{6}. \quad (4.17)$$

4.10 Competing integration methods

In this study, we compared scFLASH with ten state-of-the-art integration methods. These methods can be summarized into two categories:

1) For MBMC integration: scINSIGHT [37], scMerge2 [24], scDisInFact [56], and scDisco [25].
2) For scRNA-seq integration: BBKNN [36], Harmony [19], Scanorama [15], scVI [26], scDREAMER [46], and Seurat [50].

More details on the methods and configurations were summarized in supplementary materials (Supplementary Table 9).

4.11 Pathway enrichment analysis

The Gene Ontology (GO) analysis was performed by "enrichGO" function in clusterProfiler R package (version 4.10.1) [59], focusing on biological processes. The input was a list of DEG names obtained by the "FindMarkers" function in Seurat. The terms with false discovery rates less than 0.05 based on the Benjamin-Hochberg procedure were considered enriched pathways. For a broader pathway enrichment analysis, we also analyzed the "LINCS L1000 Ligand Perturbations up" library using the enrichR R package (version 3.2) [20]. The list of DEG names was input into the "enrichr" function, which returned the enriched terms.

4.12 Calculation of the signature scores

We built signatures using several specific gene sets (e.g. condition-specific DEGs). To calculate the signature score for each sample, we used the "AUCell_calcAUC" function in the AUCell R package (version 1.25.2) [2] with the default parameters. The inputs of the "AUCell_calcAUC" function are a ranked gene list and a gene set. In this study, the ranked gene list was obtained using the "AUCell_buildRankings" function with the normalized expression matrix as input, and the signature genes were used as the required gene set. Additionally, to quantify interferon signature expression in each cell, an interferon response score was calculated using the "tl.score_genes" tool in Scanpy, based on the type I interferon gene list (GO:0034340).

4.13 Trajectory analysis

Trajectory analysis was performed using the Palantir [42] Python package (version 1.3.3) with the scFLASH-corrected expression profiles as input. We ran Palantir using the default parameters, specifying the starting cell state as "healthy" and the terminal state as "disease". As for the dimensionality reduction and visualization, we employed the potential of heat-diffusion for affinity-based trajectory embedding (PHATE) [32] using the "external.tl.phate" function in the Scanpy package.

Data and software availability

All datasets analyzed in this study are publicly available. The corresponding descriptions and pre-processing steps can be found in the Supplementary Materials. The open-source code for scFLASH is available at GitHub: `https://github.com/SDU-Math-SunLab/scFLASH`. The Supplementary Materials, including notes, figures, and tables, is available at `https://github.com/SDU-Math-SunLab/scFLASH-supplementary`.

Author contributions

Qingbin Zhou: methodology, formal analysis, software, investigation, visualization, writing - original draft. Tao Ren: methodology, formal analysis, software, investigation. Fan Yuan: investigation. Jiating Yu: investigation. Jiacheng Leng: software, investigation. Jiahao Song: investigation. Duanchen Sun: conceptualization, methodology, investigation, supervision, funding acquisition, writing - review & editing. Ling-Yun Wu: conceptualization, methodology, investigation, supervision, funding acquisition, writing - review & editing.

These authors contributed equally: Qingbin Zhou, Tao Ren.

Acknowledgments

This work has been supported by the National Key Research and Development Program of China (Grant No. 2022YFA1004800), by the National Natural Science Foundation of China (Grant Nos. 62202269, 12231018), by the Science Foundation Program of the Shandong Province (Grant No. 2023HWYQ-012), by the Startup Foundation for Introducing Talent of Nanjing University of Information Science & Technology, China (Grant No. 2024r088), by the Open project of BGI-Shenzhen (Grant No. BGIRSZ20220005), and by the Program of Qilu Young Scholars of Shandong University.

References

[1] J. A. Agundez, F. J. Jimenez-Jimenez, H. Alonso-Navarro, and E. Garcia-Martin, *The potential of LINGO-1 as a therapeutic target for essential tremor*, Expert Opin. Ther. Targets, 19:1139–1148, 2015.

[2] S. Aibar et al., *SCENIC: Single-cell regulatory network inference and clustering*, Nat. Methods, 14:1083–1086, 2017.

[3] J. L. Andrews and F. Fernandez-Enright, *A decade from discovery to therapy: Lingo-1, the dark horse in neurological and psychiatric disorders*, Neurosci. Biobehav. Rev., 56:97–114, 2015.

[4] P. S. Arunachalam et al., *Systems biological assessment of immunity to mild versus severe COVID-19 infection in humans*, Science, 369:1210–1220, 2020.

[5] D. B. Burkhardt et al., *Quantifying the effect of experimental perturbations at single-cell resolution*, Nat. Biotechnol., 39:619–629, 2021.

[6] J. Cha and I. Lee, *Single-cell network biology for resolving cellular heterogeneity in human diseases*, Exp. Mol. Med., 52:1798–1808, 2020.

[7] E. Dann, N. C. Henderson, S. A. Teichmann, M. D. Morgan, and J. C. Marioni, *Differential abundance testing on single-cell data using k-nearest neighbor graphs*, Nat. Biotechnol., 40:245–253, 2022.

[8] C. De Donno et al., *Population-level integration of single-cell datasets enables multi-scale analysis across samples*, Nat. Methods, 20:1683–1692, 2023.

[9] Q. Duan et al., *LINCS Canvas Browser: Interactive web app to query, browse and interrogate LINCS L1000 gene expression signatures*, Nucleic Acids Res., 42:W449–460, 2014.

[10] R. M. Elgamal et al., *An integrated map of cell type-specific gene expression in pancreatic islets*, Diabetes, 72:1719–1728, 2023.

[11] Y. Ganin et al., *Domain-adversarial training of neural networks*, J. Mach. Learn. Res., 17:2096–2030, 2016.

[12] A. Goeva et al., *HiDDEN: A machine learning method for detection of disease-relevant populations in case-control single-cell transcriptomics data*, Nat. Commun., 15:9468, 2024.

[13] A. Grubman et al., *A single-cell atlas of entorhinal cortex from individuals with Alzheimer's disease reveals cell-type-specific gene expression regulation*, Nat. Neurosci., 22:2087–2097, 2019.

[14] T. Hamano, K. Hayashi, N. Shirafuji, and Y. Nakamoto, *The implications of autophagy in Alzheimer's disease*, Curr. Alzheimer Res., 15:1283–1296, 2018.

[15] B. Hie, B. Bryson, and B. Berger, *Efficient integration of heterogeneous single-cell transcriptomes using Scanorama*, Nat. Biotechnol., 37:685–691, 2019.

[16] K. H. Kaestner, A. C. Powers, A. Naji, H. Consortium, and M. A. Atkinson, *NIH initiative to improve understanding of the pancreas, islet, and autoimmunity in type 1 diabetes: The Human*

Pancreas Analysis Program (HPAP), Diabetes, 68:1394–1402, 2019.
[17] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, arXiv:1412.6980, 2014.
[18] D. P. Kingma and M. Welling, *Auto-encoding variational bayes*, arXiv:1312.6114, 2013.
[19] I. Korsunsky et al., *Fast, sensitive and accurate integration of single-cell data with Harmony*, Nat. Methods, 16:1289–1296, 2019.
[20] M. V. Kuleshov et al., *Enrichr: A comprehensive gene set enrichment analysis web server 2016 update*, Nucleic Acids Res., 44:W90–97, 2016.
[21] T. Kuret, S. Sodin-Semrl, B. Leskosek, and P. Ferk, *Single cell RNA sequencing in autoimmune inflammatory rheumatic diseases: Current applications, challenges and a step toward precision medicine*, Front. Med., 8:822804, 2021.
[22] J. S. Lee et al., *Immunophenotyping of COVID-19 and influenza highlights the role of type I interferons in development of severe COVID-19*, Sci. Immunol., 5(49):eabd1554, 2020.
[23] W. S. Liang et al., *Gene expression profiles in anatomically and functionally distinct regions of the normal aged human brain*, Physiol. Genomics, 28:311–322, 2007.
[24] Y. Lin, Y. Cao, E. Willie, E. Patrick, and J. Y. H. Yang, *Atlas-scale single-cell multi-sample multi-condition data integration using scMerge2*, Nat. Commun., 14:4272, 2023.
[25] R. Liu, K. Qian, X. He, and H. Li, *Integration of scRNA-seq data by disentangled representation learning with condition domain adaptation*, BMC Bioinformatics, 25:116, 2024.
[26] R. Lopez, J. Regier, M. B. Cole, M. I. Jordan, and N. Yosef, *Deep generative modeling for single-cell transcriptomics*, Nat. Methods, 15:1053–1058, 2018.
[27] M. D. Luecken et al., *Benchmarking atlas-level data integration in single-cell genomics*, Nat. Methods, 19:41–50, 2022.
[28] A. Ma, J. Wang, D. Xu, and Q. Ma, *Deep learning analysis of single-cell data in empowering clinical implementation*, Clin. Transl. Med., 12:e950, 2022.
[29] R. Ma, E. D. Sun, D. Donoho, and J. Zou, *Principled and interpretable alignability testing and integration of single-cell data*, Proc. Natl. Acad. Sci. USA, 121:e2313719121, 2024.
[30] M. L. Martins et al., *A potent inflammatory response is triggered in asymptomatic blood donors with recent SARS-CoV-2 infection*, Rev. Soc. Bras. Med. Trop., 55:e02392022, 2022.
[31] S. Mi et al., *LINGO-1 negatively regulates myelination by oligodendrocytes*, Nat. Neurosci., 8:745–751, 2005.
[32] K. R. Moon et al., *Visualizing structure and transitions in high-dimensional biological data*, Nat. Biotechnol., 37:1482–1492, 2019.
[33] S. Nejentsev et al., *Localization of type 1 diabetes susceptibility to the MHC class I genes HLA-B and HLA-A*, Nature, 450:887–892, 2007.
[34] E. Pairo-Castineira et al., *Genetic mechanisms of critical illness in COVID-19*, Nature, 591:92–98, 2021.
[35] Z. Piran and M. Nitzan, *SiFT: Uncovering hidden biological processes by probabilistic filtering of single-cell data*, Nat. Commun., 15:760, 2024.
[36] K. Polanski et al., *BBKNN: Fast batch alignment of single cell transcriptomes*, Bioinformatics, 36:964–965, 2020.
[37] K. Qian, S. Fu, H. Li, and W. V. Li, *scINSIGHT for interpreting single-cell gene expression from biologically heterogeneous data*, Genome Biol., 23:82, 2022.
[38] K. Rajasekhar, M. Chakrabarti, and T. Govindaraju, *Function and toxicity of amyloid beta and recent therapeutic interventions targeting amyloid beta in Alzheimer's disease*, Chem. Commun., 51:13434–13450, 2015.
[39] W. M. Rand, *Objective criteria for the evaluation of clustering methods*, J. Am. Stat. Assoc., 66: 846–850, 1971.

[40] P. S. Reel, S. Reel, E. Pearson, E. Trucco, and E. Jefferson, *Using machine learning approaches for multi-omics data analysis: A review*, Biotechnol. Adv., 49:107739, 2021.
[41] A. Sette and S. Crotty, *Adaptive immunity to SARS-CoV-2 and COVID-19*, Cell, 184:861–880, 2021.
[42] M. Setty et al., *Characterization of cell fate probabilities in single-cell data with Palantir*, Nat. Biotechnol., 37:451–460, 2019.
[43] Severe Covid-19 GWAS Group et al., *Genomewide association study of severe Covid-19 with respiratory failure*, N. Engl. J. Med., 383:1522–1534, 2020.
[44] A. K. Shalek and M. Benson, *Single-cell analyses to tailor treatments*, Sci. Transl. Med., 9:eaan4730, 2017.
[45] S. N. Shapira, A. Naji, M. A. Atkinson, A. C. Powers, and K. H. Kaestner, *Understanding islet dysfunction in type 2 diabetes through multidimensional pancreatic phenotyping: The Human Pancreas Analysis Program*, Cell Metab., 34:1906–1913, 2022.
[46] A. Shree, M. K. Pavan, and H. Zafar, *scDREAMER for atlas-level integration of single-cell datasets using deep generative model paired with adversarial classifier*, Nat. Commun., 14:7781, 2023.
[47] M. A. Skinnider et al., *Cell type prioritization in single-cell data*, Nat. Biotechnol., 39:30–34, 2021.
[48] K. Sohn, X. Yan, and H. Lee, *Learning structured output representation using deep conditional generative models*, in: Proceedings of the 28th International Conference on Neural Information Processing Systems, MIT Press, 3483–3491, 2015.
[49] E. Stephenson et al., *Single-cell multi-omics analysis of the immune response in COVID-19*, Nat. Med., 27:904–916, 2021.
[50] T. Stuart et al., *Comprehensive integration of single-cell data*, Cell, 177:1888–1902, 2019.
[51] D. Sun et al., *Identifying phenotype-associated subpopulations by integrating bulk and single-cell sequencing data*, Nat. Biotechnol., 40:527–538, 2022.
[52] W. Tang et al., *Single-cell RNA-sequencing in asthma research*, Front. Immunol., 13:988573, 2022.
[53] X. Wang et al., *Identification of distinct immune cell subsets associated with asymptomatic infection, disease severity, and viral persistence in COVID-19 patients*, Front. Immunol., 13:812514, 2022.
[54] F. A. Wolf, P. Angerer, and F. J. Theis, *SCANPY: Large-scale single-cell gene expression data analysis*, Genome Biol., 19:15, 2018.
[55] J. Zierer, C. Menni, G. Kastenmuller, and T. D. Spector, *Integration of 'omics' data in aging research: From biomarkers to systems biology*, Aging Cell, 14:933–944, 2015.
[56] Z. Zhang, X. Zhao, M. Bindra, P. Qiu, and X. Zhang, *scDisInFact: Disentangled learning for integration and prediction of multi-batch multi-condition single-cell RNA-sequencing data*, Nat. Commun., 15:912, 2024.
[57] W. Zhao et al., *Deconvolution of cell type-specific drug responses in human tumor tissue with single-cell RNA-seq*, Genome Med., 13:82, 2021.
[58] K. Xu et al., *Detecting anomalous anatomic regions in spatial transcriptomics with STANDS*, Nat. Commun., 15:8223, 2024.
[59] G. Yu, L.-G. Wang, Y. Han, and Q.-Y. He, *clusterProfiler: An R package for comparing biological themes among gene clusters*, OMICS J. Integr. Biol., 16:284–287, 2012.
[60] H. Zelova and J. Hosek, *TNF-alpha signalling and inflammation: Interactions between old acquaintances*, Inflamm. Res., 62:641–651, 2013.
[61] J. Zhao et al., *Detection of differentially abundant cell subpopulations in scRNA-seq data*, Proc. Natl. Acad. Sci. USA, 118:e2100293118, 2021.