BMC Bioinformatics

RESEARCH

Open Access

SCITUNA: single-cell data integration tool using network alignment

Aissa Houdjedj[1,2], Yacine Marouf[3], Mekan Myradov[4], Süleyman Onur Doğan[1], Burak Onur Erten[1], Oznur Tastan[4], Cesim Erten[5*] and Hilal Kazan[1*]

*Correspondence: cesim@cs.arizona.edu; hilal.kazan@antalya.edu.tr

[1] Antalya Bilim University, 07190 Antalya, Turkey
[2] Akdeniz University, 07058 Antalya, Turkey
[3] Weill Cornell Medicine, New York, USA
[4] Sabanci University, 34956 Istanbul, Turkey
[5] University of Arizona, Tucson 85721, USA

Abstract

Background: As single-cell genomics experiments increase in complexity and scale, the need to integrate multiple datasets has grown. Such integration enhances cellular feature identification by leveraging larger data volumes. However, batch effects-technical variations arising from differences in labs, times, or protocols-pose a significant challenge. Despite numerous proposed batch correction methods, many still have limitations, such as outputting only dimension-reduced data, relying on computationally intensive models, or resulting in overcorrection for batches with diverse cell type composition.

Results: We introduce a novel method for batch effect correction named SCITUNA, a Single-Cell data Integration Tool Using Network Alignment. We perform evaluations on 39 individual batches from four real datasets and a simulated dataset, which include both scRNA-seq and scATAC-seq datasets, spanning multiple organisms and tissues. A thorough comparison of existing batch correction methods using 13 metrics reveals that SCITUNA outperforms current approaches and is successful at preserving biological signals present in the original data. In particular, SCITUNA shows a better performance than the current methods in all the comparisons except for the multiple batch integration of the lung dataset where the difference is 0.004.

Conclusion: SCITUNA effectively removes batch effects while retaining the biological signals present in the data. Our extensive experiments reveal that SCITUNA will be a valuable tool for diverse integration tasks.

Keywords: Single-cell data integration, Batch effect, Rare cell types, Iterative correction

Background

Single-cell technology enables the identification of established as well as novel cell types and enhances our understanding of cell-specific molecular mechanisms [1]. While current protocols allow querying thousands of cells with a single experiment, combining data from multiple datasets further enhances the predictive power of computational methods. A major challenge in integrating multiple single-cell datasets is the presence of 'batch effects' which represent unwanted technical variation

that result from handling cells in distinct batches. These differences can stem from the use of distinct sequencing protocols, platforms, and technologies as well as from variations in sample acquisition and sample composition. Batch effects also include differences due to biological factors such as tissues, spatial locations, and donors. Mitigating these batch effects is a mandatory step before analyzing integrated data, in order to avoid unwanted technical variations that might hinder the detection of true biological signals in the data [2, 3]. A successful integration of single-cell datasets ensures that similar cells are clustered together while the local neighborhood structure of the cells and the biological signals in the original data are preserved. Additionally, cell types that are present exclusively in one dataset should remain isolated and not be merged with other cell types. Finally, the integration strategy should be robust against diverse single cell datasets, with different dropout rates and distinct cell type compositions.

Several different types of computational methods have been proposed for batch effect removal in single-cell data integration [4–6]. These methods can broadly be categorized to four groups: anchor-based, graph-based, hybrid methods that integrate both anchor-based and graph-based techniques, and methods leveraging deep learning [7].

Most of the anchor-based methods adopt the Mutual Nearest Neighbors (MNN) strategy to determine anchors [8–10]. Anchors are assumed to be cell pairs across batches that refer to the same cell type. MNN strategy identifies anchors as pairs of cells, one from each batch, that are within each other's set of k-nearest neighbors. The next step is to integrate the different batches with correction vectors calculated from the mean differences in gene expression between cells in MNN pairs. MNNCorrect applies this strategy on the original space without reducing dimensionality, and this leads to high memory consumption and runtime [8]. To alleviate this problem, several methods are proposed to identify anchors in the reduced space instead. In particular, fastMNN [8] uses PCA and Seurat MultiCCA [9] captures the most correlated pairs in low dimensional space determined with Canonical Correlation Analysis (CCA), whereas Scanorama [11] employs Singular Value Decomposition (SVD). Most of these methods perform batch correction for two batches at a time and repeat this process to integrate more than two batches. The ordering of batches can significantly affect the output. On the other hand, Scanorama identifies anchors across more than two pairs simultaneously. Harmony [12] utilizes a different strategy to identify anchors in PCA reduced space. This strategy is based on an iterative clustering algorithm where at each iteration, clusters with similar cells from diverse batches are formed by applying the soft k-means algorithm. Correction vectors are then calculated for each cluster and each batch using cluster centroids.

Graph-based methods use community detection algorithms on weighted graphs to identify shared cell populations across batches. *Conos* constructs a graph connecting cells within and across batches, using MNN for inter-batch edges and PCA for intra-batch distances [10]. Common clusters are then detected using community detection algorithms. BBKNN builds graphs by identifying k-nearest neighbors within each batch and merging them with distance metrics similar to UMAP[13]. LIGER uses integrated non-negative matrix factorization (NMF) to create a graph based on batch-specific and shared factors, followed by clustering with Louvain algorithm [14].

Deep learning-based methods have also become popular for batch effect correction. ScGen [15] uses transfer learning with variational autoencoders for dataset adjustment, but requires supervised cell type input. SAUCIE employs autoencoders to correct batch effects by minimizing reconstruction error [16]. scVI [17] combines a variational autoencoder with a Bayesian model to model observed counts, while scANVI extends scVI with semi-supervised learning using cell type labels [18–20].

Many challenges still exist in batch effect correction for single cell datasets. Most existing methods output low-dimensional representations of the original data. The absence of gene expression data for individual genes hinders downstream analysis such as differential gene expression. Methods such as scVI are computationally demanding and semi-supervised/supervised nature of scGen and scANVI hinder their robust application. Also, some methods change both the reference and the query dataset concurrently making it unsuitable for integrating user-generated datasets with standard reference data such as the Human Cell Atlas. Another key issue of batch correction methods is overcorrection where true biological signals present in the dataset can be regarded as batch effects and removed. To overcome these challenges in single cell data integration, we propose a novel graph and anchor-based method called SCITUNA, **S**ingle- **C**ell Data **I**ntegration **T**ool **U**sing **N**etwork **A**lignment. SCITUNA improves over existing anchor-based approaches in a number of ways. A novelty of SCITUNA is its use of MNN-based anchors as a basis to produce an alignment which is a many-to-one mapping between the two batches. This alignment guides the calculation of correction vector for each cell which is a combination of two terms: the difference between the cell and its aligned cell, the sum of the analogous differences for the neighbors of the cell. Another key contribution of SCITUNA is its application of an iterative procedure for integrating cells not involved in an alignment. For such cells, only neighbors contribute to the calculation of correction vectors and iterative application of these calculations enables the diffusion of information in the *network* of cells. This strategy significantly improves the integration of batches that have distinct cell type composition. Furthermore, SCITUNA outputs the integrated matrix in the original gene expression space which allows downstream applications such as differential gene expression analysis. Finally, SCITUNA uses a novel optimal transport based ordering strategy for integrating more than two batches.

We compare SCITUNA with existing state-of-the-art batch correction methods on real and simulated single-cell datasets with varying degrees of cell population and cell type differences. Performance comparisons with a diverse set of metrics show that SCITUNA performs better than the alternatives in effectively removing batch effects while retaining the biological signals present in the data. Additionally, we assess the integration of individual pairs of batches qualitatively using UMAP plots. Moreover, SCITUNA scales well to large datasets. SCITUNA is freely available at https://github.com/abucompbio/SCITUNA.

Methods

We introduce a novel method for batch effect correction named SCITUNA, a Single-Cell data Integration Tool Using Network Alignment. SCITUNA represents the intra-batch cell similarities with a graph per batch and the similarities between the batches with a bipartite graph. These graphs are utilized to produce an alignment between the

two batches. This alignment subsequently guides the integration process, transforming the expression values of cells from one batch into the expression space of another. During the transformation step, a novel iterative correction strategy is applied to those cells that do not appear in the alignment. Furthermore, integration is performed in the original gene expression space which allows downstream applications such as differential gene expression analysis. Finally, a novel ordering strategy based on optimal transport guides the ordering of the batches when more than two batches are present.

Depending on the number of cells in each batch, the batch with the smaller number of cells is set as the *query* and the other as the *reference*. In what follows we discuss in detail each component of the SCITUNA algorithm; see Fig. 1 for an overview.

Data collection and preprocessing

To evaluate our method, we utilize three scRNA-seq datasets and two scATAC-seq datasets. The datasets for the former can be listed as human *lung* dataset, human *pancreas* dataset and the mouse hindbrain development dataset. The latter is formed from peaks and windows of small mouse brain scATAC-seq dataset. We download already processed data in the form of read counts or log normalized counts for these datasets from [21] and [22]. Additionally, we employ the Splatter package [23] to simulate a dataset with two batches that each contain three cell types with proportions of 20%, 20%, and 60%. Batch effects are modeled with batch.facLoc and batch.facScale parameters that are set to 0.05 and 0.1, respectively. Dropout is applied using the "experiment" method, with dropout.shape parameter set to −0.5. These settings capture variability and sparsity typical of scRNA-seq data. Table 1 provides an overview of these datasets and the number of cells in each batch of employed datasets is shown in Supplementary Tables 1-6.

Since SCITUNA integrates two batches at a time, we split each dataset into pairs of batches, resulting in a total of 120, 36, 15, 3, 3, and 1 batch pairs for the *lung*, *pancreas*,

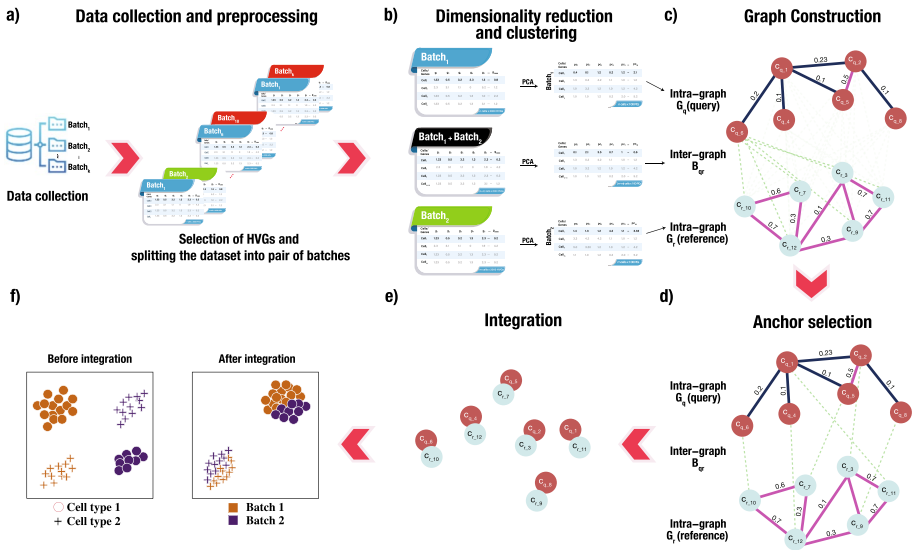

Fig. 1 The five main stages of the SCITUNA workflow: **a** preprocessing and normalization, **b** dimensionality reduction and clustering, **c** construction of intra-graphs and the inter-graph, **d** anchor selection, **e** integration, and **f** visualization of the integration results

Table 1 Statistics of employed datasets

Dataset	Batches	Cells	Features	References
Humal lung atlas (scRNA-seq)	16	32,472	15,148	[24]
Human pancreas (scRNA-seq)	9	16,382	19,093	[25–30]
Mouse hindbrain (scRNA-seq)	6	34,120	21,514	[31]
Small mouse brain (ATAC) peaks	3	11,597	94,088	[32–34]
Small mouse brain (ATAC) windows	3	10,761	110,724	[32–34]
Simulation data	2	30,000	2,000	

mouse hindbrain, *small mouse brain windows*, *small mouse brain peaks* and *simulation* datasets, respectively. The next step is to identify highly variable genes (HVGs) using scIB's "hvg_batch" function [21, 35]. After HVG selection, we obtain two matrices: D_q an $n \times g$ matrix and D_r an $m \times g$ matrix, where the subscripts q and r denote *query* and *reference* batches, respectively. Here, n and m are the numbers of cells in the query and reference batches, respectively, and g is the number of genes where $g = 2,000$. Note that exactly the same procedure is applied to peaks or windows for scATAC-seq datasets.

Dimensionality reduction and clustering

We apply Principal Component Analysis (PCA) on D_q, D_r and D_{qr} which denotes the concatenation of D_q and D_r. The number of principal components is set to 100 for each dataset [36–38]. The dimensionality reduced matrices are denoted with S_q, S_r and S_{qr}, respectively.

Next we use the k-means algorithm to cluster each dataset, S_q and S_r. To choose the number of clusters, k_c, we utilize the *silhouette index* approach. This approach evaluates the quality of clustering results by varying k_c between 2 and 30, and measuring the silhouette coefficient for each of its values. The silhouette coefficient measures how well each data point fits into its assigned cluster compared to other clusters. The silhouette coefficient ranges from -1 to 1, where a value close to 1 indicates that the data point is well-matched to its cluster and a value close to -1 indicates that the data point is likely assigned to the wrong cluster. After calculating the average silhouette coefficient values across all the data points, we select the value of k_c that results in the highest average coefficient.

Constructing the intra-graphs and the inter-graph

We construct three edge-weighted graphs: G_q, G_r, and B_{qr}. The first two are directed graphs constructed from the query and reference matrices, S_q and S_r, respectively, and are referred to as *intra-graphs*. They capture the similarity between the cells within each batch.

In what follows, we describe the construction of G_q, noting that the construction of G_r is analogous. Each node in G_q represents a cell in S_q. To determine the edges to be inserted to

G_q, Pearson correlation coefficient values are calculated between all pairs of cells in S_q. The cell pairs are then sorted based on these values, where the most positively correlated cell pairs appear at the top. We use p_q to denote the total number of edges to be inserted

to G_q and set it to the size of the smallest cluster in S_q. The cell pairs with the top Pearson correlation coefficients (5% by default) give rise to edges between corresponding node pairs in G_q without any further checks. Next, we continue to traverse the ranked list of cell pairs to insert bidirectional edges between node pairs if the corresponding cell pairs belong to the same cluster in S_q, until the total number of edges reaches the desired threshold of p_q. The edge weights are set as the Euclidean distances between the corresponding pairs of cells in S_q.

The third graph, B_{qr}, which we call the *inter-graph* is a complete undirected bipartite graph, where one partition contains nodes corresponding to the cells of the query batch and the other partition contains nodes corresponding to the cells of the reference batch. The inter-graph captures the similarity between pairs of cells from each of the two batches. Therefore, the edge weights are set as the Euclidean distance between the corresponding query cell and reference cell in the combined reduced space, that is S_{qr}.

Aligning query nodes through anchor selection

This step involves the alignment of a query node with a reference node, such that the alignment corresponds to a similar biological state across the two batches. The resulting alignments are used to integrate the query dataset with the reference dataset. To compose the alignments, we use Seurat's *anchor* selection approach [39] as a baseline. We note that Seurat's anchor selection procedure outputs a many-to-many matching. We further process this many-to-many matching to produce an *alignment* which is a many-to-one mapping. For a query cell q_i, let $M(q_i)$ be the set of reference cells mapped to q_i by the matching provided by Seurat. We assign the pair (q_i, r_j) as an alignment denoting it with the mapping $A(q_i) = r_j$ where $r_j \in M(q_i)$ is the closest matching reference cell to q_i among all cells in $M(q_i)$. The distance between each query and reference cell pair is based on the Euclidean distance calculated in the combined reduced space S_{qr}. Note that multiple query cells can be aligned with the same reference cell, whereas for a given query cell there is at most one reference cell it is aligned to.

Integrating the batch pair

The core of the SCITUNA algorithm is the integration procedure which consists mainly of four steps.

Handling isolated nodes

The intra-graphs G_q and G_r may contain isolated nodes. The number of isolated nodes depends on the number of edges inserted during the graph construction step. Since the subsequent integration process assumes that all nodes have neighbors, we add d_q closest neighbors to all isolated query nodes and d_r closest neighbors to all isolated reference nodes, where d_q, d_r correspond to the minimum degree (excluding the isolated nodes) of the query graph G_q and that of the reference graph G_r, respectively. The distance between the neighbors is defined by the Pearson correlation coefficient values between the nodes in the intra-graph. Note that we add directed edges from the isolated node to its neighbors to meet the requirements of the subsequent integration step. However, we

do not insert these edges in the opposite direction, since the neighbors of the isolated nodes may already have neighbors unless they are isolated nodes themselves.

Integration vectors of alignment nodes

In this step, for each query cell q_i that is involved in an alignment pair as defined by the mapping A, an integration vector $Iv(q_i)$ is first initialized to be the difference vector between q_i and its aligned cell $A(q_i)$ in the reference dataset. Note that this difference is calculated in the log normalized space, namely using D_q and D_r. Next, it is updated to be a convex combination of its own Iv vector and that of its neighbors in G_q with an alignment mapping. More specifically, let $N(u)$ denote the k closest neighbors of a node u in its intra-graph, where the distances are defined with respect to the edge weights in the intra-graph. $Iv(q_i)$ is updated as follows:

$$Iv(q_i) = \alpha_0\, Iv(q_i) \; + \sum_{\forall q_p \in N(q_i)} \alpha_p\, Iv(q_p)$$

Here the factors α_0, α_p are defined as:

$$\alpha_0 = \frac{1}{S}\left[\beta \; + \; (1 \; - \; \beta)\, e^{-d(q_i, A(q_i))}\right]$$
$$\alpha_p = \frac{1}{S}\left[\beta\, e^{-\frac{d(q_i,q_p)\,+\,d(A(q_i),A(q_p))}{2}} \; + \; (1-\beta)\, e^{-d(q_p, A(q_p))}\right]$$

where $S = \sum_{\forall q_p \in N(q_i)} \alpha_p$, β is a balancing parameter between the intra-graph distances and the inter-graph distances, and the distance function d stores the edge weight of the involved nodes in the relevant graph. Note that we assessed a range of values for the parameter β between 0 and 1, in increments of 0.1, and the default setting of $\beta = 0.5$ provided better performance than the other values.

Iterative Corrections on Non-Alignment Nodes

For the query nodes not involved in an alignment we employ an iterative correction procedure. For a given such node q_i, we apply the formula in Equation 2.5.2, except that now only the summation term contributes to the Iv formula. Furthermore, a neighbor q_p contributing to the formula can now be any node (aligned or not) and therefore α_p is now set to $\frac{1}{S}\left[\beta\, e^{-d(q_i,q_p)}\right]$ if q_p is a non-alignment node. Finally, a major difference between the computation of the Iv vectors of non-alignment query nodes and that of the aligned nodes is that the formula is now applied iteratively. These iterative corrections are applied until convergence, that is until there is no change in the computed vectors as compared to a previous iteration or until a maximum number of iterations is reached (default 10,000).

Final integration of the batches

In the final step of our integration algorithm, the integration vectors are used to transform the query cells, leaving reference dataset as unchanged. Each row q_i in D_q is assigned to $q_i + Iv(q_i)$. The transformed query dataset is then concatenated to the original reference dataset to produce an $(n + m) \times g$ integrated matrix.

Integration of multiple batches

We also generalize SCITUNA to enable the integration of multiple batches in addition to the integration of a pair of batches. Since SCITUNA integrates two batches at a time, an effective ordering of batches is crucial. Our strategy utilizes an iterative process in which we select the two most similar batches for integration, based on optimal cost score. Once integrated, these batches form a new composite batch and similarity scores are recalculated with the composite batch. The next iteration selects the two most similar batches again; where the batch pair to be integrated may include the composite batch or it may involve two batches that are not integrated with any other batch yet. This process continues iteratively until all batches have been integrated. To illustrate this approach, consider an example with four batches labeled A, B, C, and D. Suppose we begin by integrating batches A and B, as they are identified as the most similar pair. The resulting dataset, denoted as AB, is then used to calculate the similarity scores with batches C and D. In the next iteration, we select the pair with the highest similarity from among AB-C, AB-D, and C-D.

Let S_i and S_j denote dimensionality reduced input matrices. We define the similarity score as the optimal transport cost between S_i and S_j. Specifically, we construct a cost matrix $CM_{i,j}$ of size $(n \times m)$, where n and m represent the number of cells in S_i and S_j, respectively. This matrix consists of the Euclidean distances between each pair of cells from the two batches. Additionally, we establish a uniform distribution over the cells in both batches by initializing two probability distributions, U_i and U_j, with equal weights assigned to each cell. We then calculate the optimal transport cost using the Earth Mover's Distance [40], which quantifies the minimum cost required to transform one distribution into the other based on the cost matrix (using *emd2* from the Python package *POT: Python Optimal Transport v0.9.4*) [41].

Results

We compare SCITUNA against four of the aforementioned methods, namely Seurat, fastMNN, Scanorama, and SAUCIE. We employed the scIB-pipeline package to execute them [21]. We are unable to add Harmony, LIGER to this comparison as they provide output in the low dimensional space. Similarly, we cannot compare against BBKNN as it outputs a graph only. Lastly, scGEN and scANVI are not comparable since they require cell type labels as input.

Metrics for measuring integration accuracy

We use the metrics defined in the scIB package for the evaluations. These metrics are grouped into two main categories: *batch correction* and *biological conservation* metrics. Batch correction metrics focus on removing batch effects and include the Principal Component Regression (PCR) score, Average Silhouette Width (ASW) score (batch), graph connectivity score, and Integrated Local Inverse Simpson Index (iLISI) score. Biological conservation metrics assess the preservation of biological variance, including Normalized Mutual Information (NMI) score, Adjusted Rand Index (ARI) score, ASW (cell-type) score, Cell-type LISI (cLISI) score, isolated label F1 score, isolated label silhouette score, cell-cycle (CC) conservation score, and Highly Variable Genes (HVG)

conservation score. Note that for iLISI and cLISI scores, we have modified the scIB implementation to align it with the original R package implementation from [42]. In addition to the metrics from the scIB package, we incorporate another metric called the *over-correction score* defined in [43]. It assesses the degree of over-correction, which is calculated as the percentage of neighboring cells with inconsistent cell types among the *100* nearest neighbors. Another metric for measuring batch effects in scIB package is kBET [44]. This metric evaluates whether the local distribution of dataset labels is consistent with the global distribution. However, since kBET suffers from a significant limitation in accurately measuring batch effects when the datasets have different cell-type compositions, we have excluded it from our evaluations [12, 44, 45]. Formal definitions of all these metrics are available in Supplementary File 1.

The overall score for each method is the weighted mean of the batch correction and the bio-conservation score (as defined in [21]):

$$Score_{overall} = 0.6 \times Score_{bio} + 0.4 \times Score_{batch}$$

where $Score_{bio}$ corresponds to the average of the biological conservation metrics, and $Score_{batch}$ is the average of the batch correction metrics. Note that the weights 0.6 and 0.4 were previously used in [21] and a higher weight for bio-conservation is given to emphasize the preservation of biological signal present in the data.

Lung dataset results

We first use the *Lung* atlas to assess the performance of SCITUNA as compared to the four alternative methods. This atlas consists of three datasets for a total of 16 donors where the datasets are generated using distinct technologies and sampling techniques; see Supplementary Table 1 for a list of the number of cells in each batch. The aggregated evaluation scores are shown in Figure-2-a for all the considered methods and the scores for individual metrics are available in Table 2. SCITUNA shows the best performance with an overall score of 0.694, followed by Seurat (0.691), fastMNN (0.666), Scanorama (0.662), and SAUCIE (0.618) respectively. In terms of the aggregated biological conservation score, SCITUNA outperforms all other alternative methods. It is followed by Scanorama, Seurat, fastMNN, and SAUCIE in the order of decreasing performance. When we explore the individual metrics in the biological conservation group in more detail, we observe that SCITUNA, fastMNN, and Scanorama scores are close to each other for the majority of the metrics. SAUCIE performs the worst for all the metrics except for the isolated label silhouette score and the HVG conservation score. The difference between SAUCIE and the second-worst method is

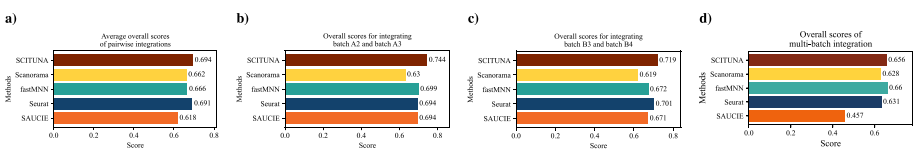

Fig. 2 Summary of the performance of SCITUNA, Scanorama, fastMNN, Seurat, and SAUCIE in terms of their overall performance scores for the *Lung* dataset. **a** Aggregated evaluation scores for 120 batch pairs within the *Lung* dataset. **b** Overall scores for integrating *A2* and *A3* batch pair. **c** Overall scores for integrating *B3* and *B4* batch pair. **d** Overall scores of multi-batch integration

Table 2 Scores of individual metrics as well as their aggregation i.e., overall score, biological conservation, and batch correction (indicated in bold) for 120 batch pairs within the Lung dataset for SCITUNA, Scanorama, fastMNN, Seurat, and SAUCIE

	Overall score	**Biological conservation**	**NMI cluster/label**	**ARI cluster/label**	**Cell type ASW**	**cLISI**	**HVG conservation**	**1-Over correction**	**CC conservation**	**Isolated label F1**	**Isolated label silhouette**	**Batch correction**	**Graph connectivity**	**iLISI**	**Batch ASW**	**PCR batch**
SCITUNA	**0.694**	**0.707**	0.772	0.694	0.579	0.967	0.687	0.842	0.801	0.493	0.524	**0.676**	0.928	0.243	0.807	0.727
Scanorama	**0.662**	**0.691**	0.777	0.712	0.591	0.963	0.53	0.837	0.79	0.486	0.53	**0.62**	0.914	0.172	0.739	0.655
fastMNN	**0.666**	**0.67**	0.785	0.71	0.589	0.985	0.323	0.843	0.771	0.498	0.527	**0.661**	0.934	0.212	0.818	0.681
Seurat	**0.691**	**0.676**	0.739	0.645	0.556	0.958	0.665	0.807	0.733	0.455	0.523	**0.713**	0.922	0.289	0.812	0.827
SAUCIE	**0.618**	**0.548**	0.512	0.343	0.51	0.94	0.484	0.595	0.671	0.343	0.539	**0.722**	0.767	0.56	0.618	0.946

The bold metrics correspond to aggregated metrics

substantial for most metrics, such as a difference of 0.343 and 0.645, respectively, for the ARI cluster/label score. We also observe notable differences between the HVG conservation scores obtained by the various methods. The highest HVG conservation score is obtained by SCITUNA which is followed by Seurat. The other methods perform significantly worse and they rank as follows from highest to lowest: Scanorama, SAUCIE, and fastMNN.

When we assess the performance in terms of batch correction metrics, we observe that SAUCIE is the top performing method which is followed by Seurat, SCITUNA, fastMNN and Scanorama. SAUCIE's performance varies significantly across the individual batch correction metrics; it ranks the worst in terms of batch ASW and graph connectivity, whereas it performs the best in terms of PCR batch score and iLISI. In particular, iLISI scores of SAUCIE and the second best method are 0.560 and 0.289 indicating a large difference. SAUCIE's extreme performance for iLISI together with its poor ranking in terms of biological conservation score indicates that it overcorrects the batch effects at the expense of poor biological effect conservation. On the other hand SCITUNA's competitive batch correction score together with its top performance in biological conservation score indicates its ability to balance these two objectives.

In Fig. 2-b and c, we present the results for two specific batch pairs (see Supplementary File 2 for the results of other batch pairs); the most similar and the least similar batch pairs, where *similarity* is based on gene expression profiles of the batches. Specifically, for each cell type, we calculate the pairwise cosine similarity scores between cell pairs belonging to that specific cell type from each batch. The similarity score for a cell type is then determined as the average of these pairwise cosine similarities. Finally, the overall similarity score for the batch pair is computed as the average of the similarity scores across all cell types. Figure 2-b contains the results for A2 (1,454 cells) and A3 (1,226 cells) batch pair which demonstrate the highest similarity score. We observe that SCITUNA outperforms all the alternatives in terms of overall score with a large margin. Namely, the ranking of the method from best to worst is as follows: SCITUNA (0.744), fastMNN (0.699), Seurat (0.694), SAUCIE (0.694), and Scanorama (0.630). Regarding the biological conservation score, SCITUNA slightly trails behind Seurat, with a marginal difference of 0.002. In terms of batch correction score, SCITUNA shows the top performance with a s core of 0.781, followed by SAUCIE (0.775), fastMNN (0.772), Seurat (0.653), and Scanorama (0.576).

Figure 2-c shows the results of the integration of batch B3 (1,911 cells) and B4 (2,353 cells) pair which have the the lowest similarity score. We observe that SCITUNA outperforms all the alternatives in terms of overall score, aggregated biological conservation score, and aggregated batch correction score. SCITUNA's superior performance compared to SAUCIE in terms of the batch correction metrics is notable, particularly considering SAUCIE's tendency to overcorrect for batch effects. Among biological conservation metrics, Scanorama, fastMNN, and SAUCIE perform poorly in terms of the HVG conservation score. All the methods result in lower scores for the metrics within the biological conservation group for integrating the B3-B4 batch pair as compared to integrating the A2-A3 batch pair (see Supplementary File 2 for more details). This is expected due to the low similarity between the two batches. An

exception is the isolated label F1 score where the scores for the A2-A3 batch pair are particularly low. We confirm that this is due to the existence of very few isolated cells in A2-A3 integration. All the methods struggle to accurately separate these cells from other types of cells.

Figure 2-d shows the results of integrating multiple batches within the *Lung* atlas dataset. The results indicate that fastMNN and SCITUNA are closely ranked, both achieving high scores of 0.66 and 0.656, respectively. They are followed by Seurat (0.631), Scanorama (0.628), and SAUCIE (0.457). In terms of the aggregated biological conservation score, SCITUNA slightly trails behind Scanorama, with a marginal difference of 0.005, followed by fastMNN, Seurat, and SAUCIE. Specifically, SCITUNA demonstrates strong performance in most of the metrics in this category. In terms of HVG and cell cycle conservation, SCITUNA achieves scores of 0.516 and 0.826, respectively. Scanorama is ranked as the second-best method for these metrics, with scores of 0.384 and 0.771, respectively. We observe that SAUCIE, Seurat, and fastMNN over-correct the data during batch mixing, resulting in a loss of biological information. This issue is reflected in their batch correction scores, where both Seurat and fastMNN exhibit high batch correction scores while still demonstrating low biological conservation scores. In contrast, SCITUNA performs very closely to these methods while effectively preserving relevant biological information despite batch effects. As shown in Supplementary Table 1, the *Lung* dataset contains groups of batches with highly similar cell type composition, while significant differences exist between the cell type compositions across these groups. To demonstrate that SCITUNA's superior performance is not solely attributed to the

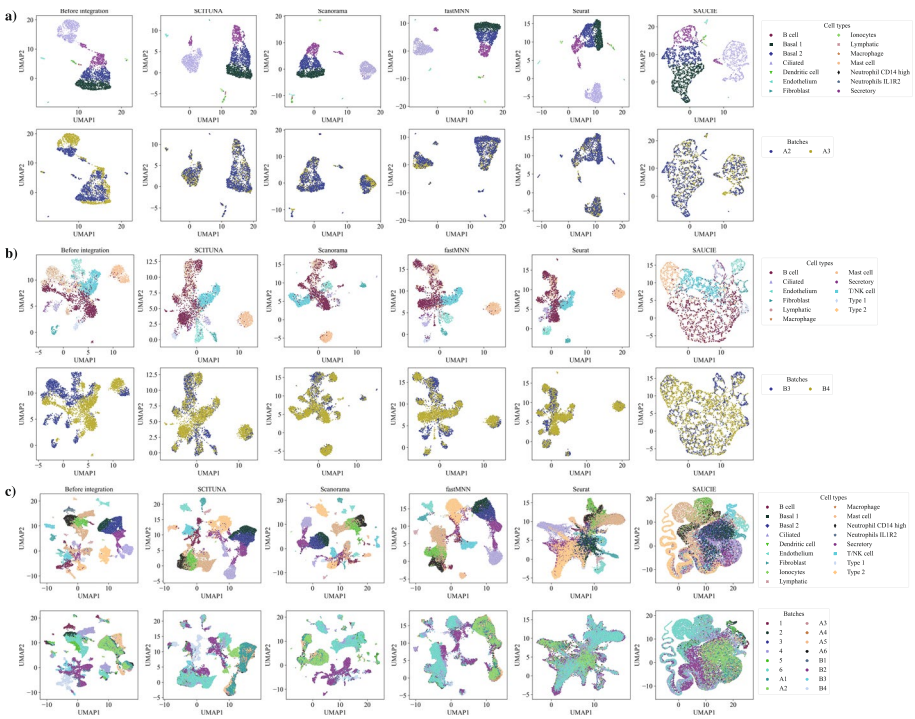

Fig. 3 UMAP plots of the *Lung* dataset: **a** A2-A3, **b** B3-B4 batches, and **c** multi-batch integration before and after integration using SCITUNA, Scanorama, fastMNN, Seurat, and SAUCIE. Each subfigure is labeled according to cell type identities in the first row and batch identities in the second row

integration of batches with diverse cell type compositions, we repeat our experiments using a subset of batches with similar cell type compositions (i.e., A1-A2-A3-A4-A5). Supplementary Figure 1 presents the results, which show that SCITUNA continues to achieve top performance, indicating its robust efficacy in this scenario as well.

Figures 3-a and b show the UMAP plots illustrating the original data and the outputs of integration methods for the batch pair A2-A3 and the batch pair B3-B4, respectively (see Supplementary Figures 2-121 for the UMAP plots of other batch pairs). For the batch pair A2-A3, we observe that SCITUNA and Seurat outperform the other methods regarding batch mixing. On the other hand, Scanorama and fastMNN show insufficient mixing of batches within the cluster of ciliated cells. Scanorama also struggles to effectively mix endothelial cells from both batches. Unlike other methods, SAUCIE produces large loose clusters with inadequate batch mixing for a subset of ciliated cells. In terms of cell types, we observe that Basal 1 and Basal 2 cell types are consistently close to each other, as expected. Additionally, the proximity of ciliated cells to basal cells aligns with the known biological phenomenon of basal cells differentiating into secretory cells. All the methods struggle in separating immune cell types from each other i.e., macrophages, dendritic cells, mast cells, and neutrophils. This difficulty is expected due to the relatively small number of cells from each immune cell type. Among all the methods, Scanorama is the only method which mixes endothelial cells with immune cells. Interestingly, SCITUNA positions ionocytes close to the cluster of immune cell types diverging from other methods which separate ionocytes from other cell types. Another observation relates to a subset of ciliated cells situated close to secretory cells. Remarkably, this subpopulation comprises cells from both batches, indicating distinctive characteristics confirmed by both data sources. This subpopulation is located closer to the cluster of secretory cells across the outputs of the majority of the integration methods. Another notable observation is that, in contrast to other methods, SAUCIE places immune cell types near the large clusters containing secretory and basal cells.

For the B3-B4 batch pair, we observe that SCITUNA produces clusters that evenly mix cells from different batches. The other methods perform similarly with the exception of fastMNN which shows difficulty in mixing a subset of cells from B3 with the other batch. In terms of cell types, all the methods struggle in distinguishing the large number of B cells from other cell types including macrophages, type 2, ciliated, and lymphatic cells. Notably, Seurat positions a small set of B cells from B3 batch apart from the main cluster of B cells. Additionally, the UMAP plot of the original dataset reveals a subpopulation of fibroblast cell type which only contains cells from B3 batch. Scanorama and Seurat position this subpopulation separately from the main cluster of fibroblasts, whereas SCITUNA and fastMNN are better at integrating it to the main fibroblast cluster. Another notable difference arises with Scanorama which separates two small subpopulations of ciliated cells from the main ciliated cell cluster. Furthermore, all the methods have similar difficulties in accurately clustering secretory cells apart from other cell types, indicating a common challenge in discerning this particular cell type.

Figure 3-c shows the UMAP plots illustrating the original data and the outputs of integration methods for multiple batch integration. We observe that SCITUNA and Scanorama accurately separate cell types while mixing the batches. In terms of cell types, we observe that endothelial cells form multiple groups in the unintegrated dataset,

indicating a high level of heterogeneity within this cell type. In contrast, SCITUNA creates a compact and distinct cluster for endothelial cells, effectively grouping the majority of them together. Conversely, the other methods either separate these cells into distinct groups or mix them with other cell types. Namely, Scanorama separates endothelial cells into two distinct clusters, while fastMNN and Seurat fail to separate them from fibroblasts or lymphatic cells, respectively. Similarly, SCITUNA distinguishes itself from other methods by effectively clustering mast cells. Scanorama separates mast cells into two distinct groups whereas Seurat and SAUCIE merge them with other cell types. Furthermore, neutrophils (CD14 high), dendritic cells, and macrophages cluster closely together, forming dense and distinct groups in SCITUNA, Scanorama, and fastMNN, whereas Seurat produces larger and more dispersed groupings. SCITUNA also creates a distinct and well-separated cluster for T/NK cells, with minimal mixing with other cell types. In contrast, Scanorama separates them into two different groups whereas Seurat and SAUCIE mix these cells with other immune cells. Fibroblast cells form three distinct groups in the original dataset. SCITUNA and fastMNN succeed in grouping most of these cells together, separating them from other cell types. In contrast, Scanorama clusters them into three distinct groups, whereas Seurat's integration results in less defined boundaries between fibroblast, macrophage and B cells. Lastly, as mentioned before, SAUCIE demonstrates poor performance in both mixing the batches and separating the cell types effectively.

Pancreas dataset results

We next use the *Pancreatic* dataset to assess SCITUNA's performance; see Supplementary Table 2 for a list of the number of cells in each batch. The evaluation scores for SCITUNA and the other four methods are shown in Figure-4-a. SCITUNA shows the best performance with an overall score of 0.743, followed by Seurat (0.736), fastMNN (0.710), SAUCIE (0.709), and Scanorama (0.693). The scores for individual metrics across all the methods are available in Supplementary File 2. In terms of biological conservation, SCITUNA outperforms other methods, leading with a score of 0.734. The ranking of the other methods from best to worst is as follows: Seurat (0.728), Scanorama (0.713), fastMNN (0.682), and SAUCIE (0.671). If we consider individual metrics within the biological conservation category, we observe that SCITUNA, fastMNN, Scanorama, and Seurat scores are close to each other for the majority of the metrics except for the CC conservation and the HVG conservation scores. For these two metrics, Scanorama and fastMNN perform much worse than the other methods. Additionally, SAUCIE's performance diverges from the other four methods significantly, scoring notably lower in metrics such

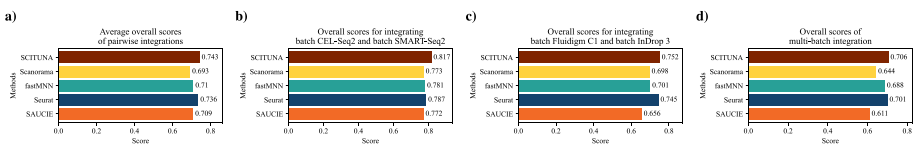

Fig. 4 Summary of the performance of SCITUNA, Scanorama, fastMNN, Seurat, and SAUCIE in terms of their overall performance scores for the *Pancreatic* dataset. **a** Aggregated evaluation scores for 36 batch pairs within the *Pancreatic* dataset. **b** Overall scores for integrating CEL-Seq2 and SMART-Seq2 batch pair. **c** Overall scores for integrating Fluidigm C1 and inDrop3 batch pair. **d** Overall scores of multi-batch integration

as NMI cluster/label, ARI cluster/label, isolated label F1, and the overcorrection score. On the other hand, in terms of cell type ASW, and CC conservation scores, SAUCIE exhibits superior performance compared to the other methods though the difference from the second-ranking method is marginal.

Scanorama performs significantly worse than the other methods in terms of aggregate batch correction score. The rankings for each individual metric in this group vary. Namely, the top method for Batch ASW and PCR batch metric are SCITUNA and Seurat, respectively. Notably, iLISI reveals substantial differences among the methods, with SAUCIE achieving the highest score of 0.546, followed by fastMNN at 0.31 and SCITUNA at 0.303. In contrast, Scanorama, which performs the worst, only achieves a score of 0.071.

Similar to the lung dataset, we show detailed results for two selected batch pairs. CEL-Seq2 (2,285 cells) and SMART-Seq2 (2,394 cells) are determined as the most similar batch pair, and Fluidigm C1 (638 cells) and inDrop3 (3,605 cells) are determined as the most dissimilar batch pair. Note that the similar and the dissimilar pairs are determined using the same procedure employed for the *Lung* dataset. The aggregated scores for both pairs of batches are shown in Figure-4-b and Figure-4-c, respectively (see Supplementary File 2 for the results of other batch pairs). For CEL-Seq2 and SMART-Seq2 batches, we observe that SCITUNA outperforms the other methods with respect to the overall score (0.817) and the batch correction score (0.863). The rankings of the rest of the methods are as follows from best to worst in terms of overall score: Seurat (0.787), fastMNN (0.781), Scanorama (0.773), and SAUCIE (0.772). In terms of the batch correction score they are ranked as follows: fastMNN (0.857), SAUCIE (0.848), Seurat (0.785), and Scanorama (0.775). In terms of the biological conservation scores, Seurat outperforms SCITUNA with a marginal difference of 0.003. The ranking of the methods from the best to the worst is as follows: Seurat (0.789), SCITUNA (0.786), Scanorama (0.771), fastMNN (0.73), and SAUCIE (0.721).

For Fluidigm C1 and inDrop3 batches, SCITUNA again has the best overall score (0.752) and batch correction score (0.773), followed by Seurat (0.745, 0.751), fastMNN (0.701, 0.715), Scanorama (0.698, 0.684), and SAUCIE (0.656, 0.683). In terms of the biological conservation score, the ranking of the methods is as: Seurat (0.741), SCITUNA (0.738), Scanorama (0.708), fastMNN (0.692), and SAUCIE (0.637). To summarize, SCITUNA's top overall performance in integrating this batch pair indicates that its integration strategy preserves biological information without over-correcting batch effects.

An interesting comparison can be made between the scores obtained for the integration of CEL-Seq2 - SMART-Seq2 batch pair and Fluidigm C1 - inDrop3 batch pair. We observe that the methods perform dramatically worse in terms of isolated label F1 score and iLISI score for the latter batch pair which is in line with the fact that these two batches are quite diverse.

Figure 4-d depicts the results of integrating multiple batches within the *Pancreatic* dataset. The results indicate that SCITUNA achieves the top overall score (0.706). The ranking of the other methods from best to worst is as follows: Seurat (0.701), fastMNN (0.688), Scanorama (0.644), and SAUCIE (0.611). Additionally, SCITUNA achieves the top aggregated biological conservation score of 0.695, followed by Seurat (0.688), fastMNN (0.655), Scanorama (0.620), and SAUCIE (0.591). When considering the

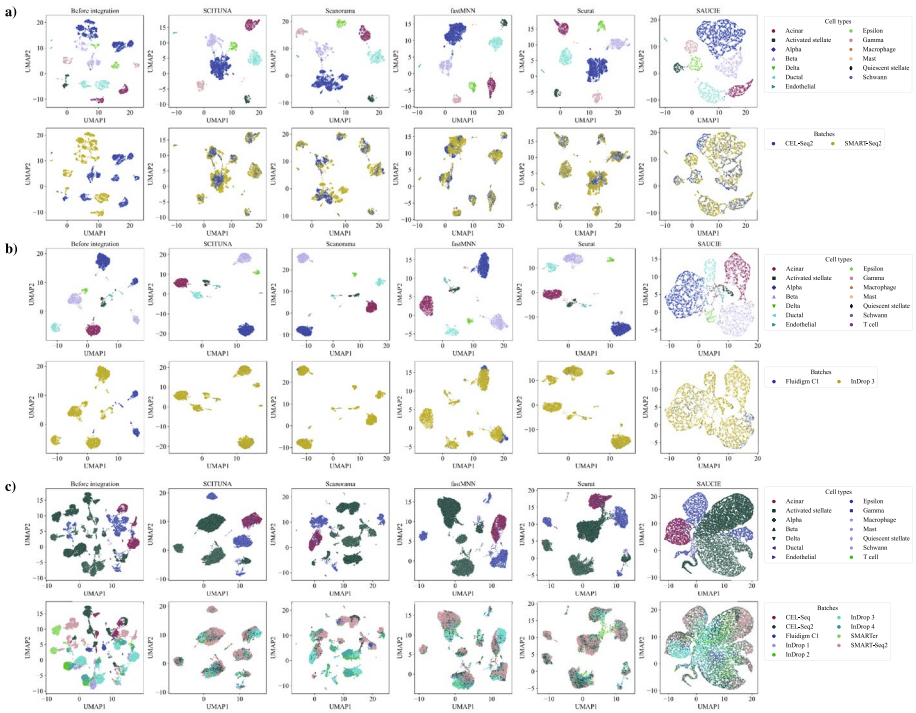

Fig. 5 UMAP plots of the *Pancreatic* dataset: **a** CEL-Seq2 and SMART-Seq2, **b** Fluidigm C1 and CEL-Seq2 batches, **c** multi-batch integration before and after integration using SCITUNA, Scanorama, fastMNN, Seurat, and SAUCIE. Each subfigure is labeled according to cell type identities in the first row and batch identities in the second row

aggregated batch score, we observe that fastMNN leads with a marginal difference of 0.014, followed closely by SCITUNA and Seurat. In contrast, while SAUCIE shows the highest iLISI score, it also has the lowest aggregated batch correction score and exhibits issues with over-correction.

UMAP visualization of the original data in Fig. 5-a shows that the cell types from each batch are located far apart indicating strong batch effects. On the other hand, most of the cell types have clear boundaries indicating homogeneity within each type. All the methods except SAUCIE are able to integrate the two batches effectively for the majority of the cell types. Scanorama, fastMNN, and SAUCIE positions a subset of ductal cells close to the cluster of acinar cells, whereas SCITUNA and Seurat are able to effectively cluster these cells correctly. Additionally, SCITUNA shows a better performance in separating alpha cells from the other cells. Methods also diverge in their placement of epsilon and gamma cells. SCITUNA and Seurat position these two cell types close to each other though within two distinct clusters. On the other hand, Scanorama and fastMNN position epsilon cells far apart from gamma cells. In Scanorama's outputs, epsilon cells are close to beta cells, while in fastMNN's outputs, they are close to alpha cells. Additionally, despite the small number of macrophage and mast cells, all methods effectively separate these two cell types from others.

Figure 5-b shows the integration of Fluidigm C1 and inDrop3 batches. This serves as a challenging integration example due to the difference in cell types as well as the difference in the number of total cells. In terms of batch mixing, fastMNN performs

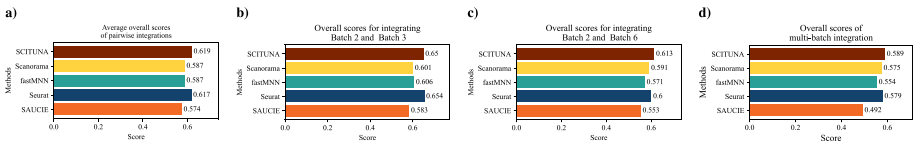

Fig. 6 Summary of the performance of SCITUNA, Scanorama, fastMNN, Seurat, and SAUCIE in terms of their overall performance scores for the *Mouse hindbrain* dataset. **a** Aggregated evaluation scores for 15 batch pairs within the *Mouse hindbrain* dataset. **b** Overall scores for integrating Batch 2 and Batch 3. **c** Overall scores for integrating Batch 2 and Batch 6. **d** Overall scores of multi-batch integration

worse than the other methods where we are able to observe that the batches remain distinct from each other in certain parts of the UMAP plot. SCITUNA consistently excels in clustering cells from the same cell type. It effectively mixes batches while ensuring clear separation between different cell types. In particular, SCITUNA is better at separating endothelial cells from other cell types whereas Scanorama and fastMNN mixes it with other cell types such as schwann, activated, and quiescent stellate cells. Similarly, SCITUNA succeeds in separating gamma cells from other cells, whereas Scanorama mixes them with alpha and epsilon cells. See Supplementary Figures 122-157 for the UMAP plots of other batch pairs.

Figure 5-c depicts the integration of multiple batches within the Pancreatic dataset. In terms of cell types, SCITUNA consistently demonstrates the best separation and compact clustering across all the cell types, preserving clear boundaries and minimizing batch effects. In particular, SCITUNA generates a compact and well-defined cluster for gamma cells, exhibiting minimal overlap with other cell types. In contrast, other methods demonstrate some overlap between the clusters of neighboring cell types. Additionally, SCITUNA, fastMNN, and Seurat create tight and distinct groups for delta (or beta) cells whereas Scanorama splits these cells into two distinct groups. Notably, fastMNN uniquely positions endothelial cells at a significant distance from both activated and quiescent stellate cells. In contrast, SCITUNA, Scanorama, and Seurat group these cell types into closely positioned clusters. Similar to the other datasets, SAUCIE generally performs the worst, with diffuse clusters and significant overlap between cell types.

Mouse hindbrain developmental dataset results

In addition to the analysis of datasets from human, we use the *Mouse hindbrain* dataset to evaluate the performance of SCITUNA compared to other alternatives. This dataset includes six different batches; See Supplementary Table 3 for the distribution of cell types in each batch. Figure 6-a depicts the aggregated evaluation scores for all considered methods. SCITUNA achieves the highest overall score among all methods, with a score of 0.619, followed by Seurat (0.617), Scanorama (0.587), fastMNN (0.587), and SAUCIE (0.574). The scores for individual metrics across all the methods are available in Supplementary File 2. In terms of biological conservation, SCITUNA demonstrates the top performance with an overall score of 0.626. The other methods rank as follows, from highest to lowest: Seurat (0.601), Scanorama (0.600), fastMNN (0.566), and SAUCIE (0.500). When examining individual metrics within

the biological conservation category, SCITUNA consistently outperforms the other methods. In contrast, SAUCIE exhibits significant challenges, notably over-correcting the dataset and losing important biological information, followed by Seurat, Scanorama, and fastMNN.

As it is done for other datasets, we present detailed results for two selected batch pairs from the *Mouse hindbrain* data. Figure 6-b illustrates the integration results for Batch 2 (5,496 cells) and Batch 3 (5,103 cells), identified as the most similar batch pair. SCITUNA and Seurat show close overall scores of 0.650 and 0.654, respectively. This is followed by fastMNN (0.606), Scanorama (0.601), and SAUCIE (0.583). In terms of the aggregated biological conservation score, SCITUNA outperforms the other methods with a score of 0.623. When considering the batch correction score, Seurat achieves a significantly high score of 0.74, while the second-ranked method's score is 0.692. However, SAUCIE over-corrects the dataset notably, resulting in significant loss of biological information. Figure 6-c shows the integration results for Batch 2 (5,496 cells) and Batch 6 (3,730 cells), chosen as the most dissimilar batch pair. The results show that SCITUNA achieves an overall score of 0.613, followed by Seurat (0.6), Scanorama (0.591), fastMNN (0.571), and SAUCIE (0.553). With regards to biological conservation score, SCITUNA outperforms the other methods and excels in most of the metrics within this category. The *Mouse hindbrain* dataset contains 53 distinct cell types, making it challenging to visualize individual clusters in the UMAP plots. See Supplementary Figures 158-172 for the UMAP plots of other batch pairs.

Figure 6-d shows the results of integrating multiple batches within the *Mouse hindbrain* dataset. According to the results, SCITUNA shows a high overall score of 0.589, followed by Seurat (0.579), Scanorama (0.575), fastMNN (0.554) and SAUCIE (0.492). In terms of the biological conservation scores, SCITUNA outperforms the other method with a score of 0.569. The ranking of the other methods are as follows: Scanorama (0.559), Seurat (0.52), fastMNN (0.513), and SAUCIE (0.421). In terms of batch correction scores, Seurat demonstrates superior performance, leading by a margin of 0.05 over SCITUNA. However, it also faces challenges with overcorrection. This underscores the need for careful consideration in balancing effective batch correction with the preservation of biological variability. The UMAP plot showing the integration of multiple batches within the *Mouse hindbrain* dataset is provided in Supplementary Figure 173.

Small mouse brain (ATAC) dataset results

To assess the robustness of SCITUNA for datasets other than scRNA-seq, we also provide results on scATAC-seq dataset. Specifically, we focus on the *Small mouse brain (ATAC) windows* and *Small mouse brain (ATAC) peaks*, each consisting of datasets from three distinct sources. The first source is a dataset from fresh cortex tissue of an adult mouse brain (P50), retrieved from 10x Genomics [33]. The second source includes a dataset collected from 8-week-old male C57BL/6J mice [32]. Finally, the third dataset consists of six samples from [34]. The results for the *Small mouse brain (ATAC) windows* are presented in the main manuscript, while the findings for the *Small mouse brain (ATAC) peaks* are available in the Supplementary. The cell type composition in each batch is available in Supplementary Tables 4-5. For both datasets, we employ the same strategy in selecting HVGs in the scRNA-seq dataset to reduce

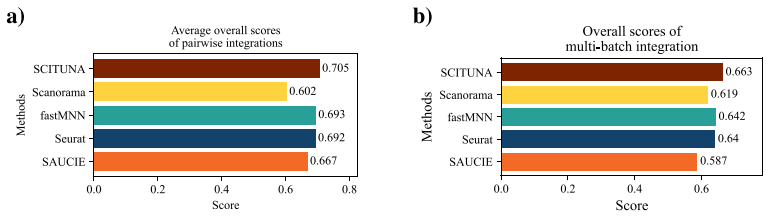

Fig. 7 Summary of the performance of SCITUNA, Scanorama, fastMNN, Seurat, and SAUCIE in terms of their overall performance scores for the *Small mouse brain (ATAC) windows* dataset. **a** Aggregated evaluation scores for 3 batch pairs within the *Small mouse brain (ATAC) windows* dataset. **b** Overall scores of multi-batch integration

noise and sparsity. In this case, we specifically select 2,000 windows (or peaks) with the highest variability. Unlike scRNA-seq data, which directly reflects transcriptional activity and cell cycle states, ATAC-seq data primarily provides insights into regulatory elements and chromatin accessibility. Consequently, following [21] we exclude CC conservation and HVG conservation metrics from our evaluation metrics, as they are not relevant for assessing the biological significance of ATAC-seq data.

Figure 7-a depicts the average overall scores for the pairwise integration of the datasets. The results show that SCITUNA achieves the highest overall score of 0.705. The rankings of the other methods, from best to worst, are as follows: fastMNN (0.693), Seurat (0.692), SAUCIE (0.667), and Scanorama (0.602). In terms of aggregate biological conservation scores, SCITUNA also leads with a score of 0.684, followed by Seurat (0.643), fastMNN (0.641), Scanorama (0.621), and SAUCIE (0.567). When examining the metrics specifically related to batch correction, SAUCIE demonstrates the highest iLISI score; however, it tends to over-correct the dataset, leading to batch mixing that obscures biological insights. In contrast, SCITUNA and Seurat achieve scores that are close to each other, with marginal differences. SCITUNA stands out by effectively balancing batch mixing and preserving biological information. Since this dataset consists of only three batches, we provide the results for individual batch pairs in Supplementary File 2.

Figure 7-b shows the overall scores from the multiple batch integration for all employed methods. SCITUNA demonstrates the highest overall score of 0.663, followed by fastMNN (0.642), Seurat (0.640), Scanorama (0.619), and SAUCIE (0.587). In terms of aggregated biological conservation scores, Scanorama and Seurat slightly outperform SCITUNA, with differences of 0.03 and 0.007, respectively. When considering the individual metrics within this category, SCITUNA shows the best performance for NMI, ARI, and cLISI scores. Additionally, in terms of over-correction metrics, SCITUNA stands out by successfully preventing over-correction of the dataset while still attaining the highest aggregated batch correction score. As a result, SCITUNA ranks at the top, followed by SAUCIE, fastMNN, Seurat, and Scanorama. These findings underscore SCITUNA's capability to integrate multiple batches effectively while preserving the biological relevance of the data. The UMAP plots for the small mouse brain windows and the small mouse brain peaks datasets are provided in Supplementary Figures 174-182. Detailed metrics for both datasets are available in Supplementary File 2.

Results on simulation dataset

To further evaluate the robustness of SCITUNA, we include results on a simulated dataset. The dataset consists of two batches with distinct cell type compositions, as detailed in Supplementary Table 6. To evaluate the performance of the methods, we exclude cell cycle conservation from our evaluation metrics. Supplementary Figure 183 presents the scores for individual metrics as well as their aggregation i.e., overall score, biological conservation, and batch correction. SCITUNA achieves the highest overall score (0.879) and biological conservation score (0.877), followed by Scanorama (0.866, 0.875), Seurat (0.814, 0.874), fastMNN (0.812, 0.821), and SAUCIE (0.827, 0.755). Notably, within the biological conservation category, SCITUNA demonstrates a superior performance for the HVG conservation score. Although SAUCIE achieves the best batch correction score, followed by SCITUNA, it suffers from overcorrection, which leads to a loss of biological information compared to SCITUNA.

Scalability

We evaluate the runtime and memory requirements of SCITUNA in comparison to other methods for all the employed datasets. Supplementary Figure 184 shows the peak memory usage and running time for pairwise integrations where average values across all pairwise integrations are reported for each dataset. We observe that the ordering of the methods in terms of runtime and memory usage depends significantly on the dataset. SAUCIE takes longer than the other methods for Human lung and pancreas datasets whereas Seurat takes longer than the other methods for Mouse Hindbrain dataset. On the other hand, SCITUNA is slower than the other methods for scATAC-seq data indicating that the larger number of features in this dataset particularly affects its running time. Overall, SCITUNA is faster than some of the other compared methods for pairwise integrations. SCITUNA requires more memory than other methods for all the datasets except Mouse Hindbrain. On the other hand, the peak memory usage of SCITUNA is always less than 12 GBs, which is not considered large by current computational standards.

Similarly, Supplementary Figure 185 shows the peak memory usage and running time of all the methods for multi-batch integration. Seurat requires the most memory for the Human Pancreas and Mouse Hindbrain datasets, while SCITUNA has the highest memory requirement for the Human Lung and scATAC-seq datasets. In terms of runtime, SCITUNA takes longer than the other methods for all the datasets. We observe that the running time of SCITUNA is highly sensitive to the ordering of the batches in multiple batch integration. For instance, in the Human Lung dataset, the chosen ordering results in one batch becoming very large, negatively impacting both time and memory usage. We also observe that the running times of Seurat and SCITUNA are comparable. This is expected as SCITUNA utilizes the anchor selection procedure of Seurat which is time and memory intensive. The incompatibility between SCITUNA (Python) and Seurat (R) also increases the total runtime, as large data has to be transferred between the two environments.

Conclusion

Integrating single-cell data is a challenging problem that requires merging data from different batches, while keeping similar cells separate and preserving the local structure of the cells. We present SCITUNA, a novel single-cell data integration approach that combines both graph-based and anchor-based techniques. SCITUNA constructs a graph for each batch to represent intra-batch cell similarities, and a bipartite graph to capture inter-batch similarities. This transforms the integration problem into a many-to-one matching problem, where cells from a query batch are matched with cells from a reference batch. The resulting matches are then used to transform the query cell space to the reference cell space. A key contribution of SCITUNA is its iterative correction strategy, which addresses unmatched cells by considering the intra-graphs of local neighborhoods to preserve the local structure within the query batch. Additionally, SCITUNA operates directly in the original gene expression space, which facilitates downstream analyses such as differential gene expression. The method also introduces a novel batch ordering strategy based on optimal transport cost, leading to improved integration results.

SCITUNA is evaluated against four well-known single-cell integration methods: Seurat, Scanorama, fastMNN, and SAUCIE on three different scRNA-seq datasets (*Human lung atlas*, *Human pancreas*, and *Mouse hindbrain*), a scATAC-seq dataset (*Small mouse windows/peaks*), and a *simulation* dataset using a number of metrics previously utilized in a benchmark study [21]. Results demonstrate that SCITUNA outperforms state-of-the-art methods, achieving a better aggregate overall score that balances integration efficiency with biological conservation. Notably, SCITUNA excels in biological conservation, with significantly higher scores compared to other methods. This is crucial, as a common issue known as overcorrection occurs when methods overemphasize dataset integration at the expense of preserving the intrinsic biological structure. One of the key contributions of SCITUNA is the ability to perform correction in a balanced way for scRNA-seq, scATAC-seq, and simulation datasets that span multiple organisms.

SCITUNA utilizes an anchor selection procedure based on finding MNNs from two different batches. When the batches share only a small subset of cell types, many cells may not have a matching anchor. This limitation is partially addressed by SCITUNA's integration procedure, which helps mitigate the impact of missing anchors. Because SCITUNA leverages Seurat's anchor selection procedure to determine anchors between batches, it has a runtime comparable to Seurat for pairwise integrations. For multiple batch integration, SCITUNA takes longer than Seurat, as it often selects an ordering that results in one batch becoming particularly large. In the future, we plan to optimize the anchor selection process to improve runtime. On the other hand, for cases where the dataset contains fewer than 50,000 cells and integration accuracy is critical, SCITUNA remains the preferred method.

Another limitation of the current study is that SCITUNA has been tested on human and mouse datasets only. Future experiments will expand these analyses to include other organisms. Additionally, we plan to extend SCITUNA to integrate spatial single-cell datasets, where the inclusion of spatial location information will be a key consideration.

Abbreviations

SVD Singular value decomposition
MNN Mutual nearest neighbors

HVGs	Highly variable genes
PCA	Principal component analysis
CCA	Canonical correlation analysis
NMF	Non-negative matrix factorization
PCR	Principal component regression
ASW	Average silhouette width
iLISI	Integrated local inverse simpson index
ARI	Adjusted rand index
cLISI	Cell-type local inverse simpson index
CC	Cell-cycle

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06087-3.

Supplementary file 1.

Supplementary file 2.

Author contributions

CE and HK designed the overall framework. CE, AH, YM, and HK contributed equally to the algorithm design. YM and AH implemented the algorithm. AH implemented the evaluation scripts. MM and OT executed experiments on dimensionality reduction step. SOD and BOE assisted with data analysis. CE, HK, and AH wrote the manuscript with input from all the authors. All authors read and approved the final manuscript.

Funding

This work has been supported by the Scientific and Technological Research Council of Turkey [121E491].

Availability of data and materials

SCITUNA and the simulation dataset are available at https://github.com/abu-compbio/SCITUNA. The processed versions of the ***Lung***, ***Pancreas***, ***Small mouse brain (ATAC) windows***, and ***Small mouse brain (ATAC) peaks*** datasets can be accessed at https://doi.org/10.6084/m9.figshare.12420968.v8 [21]. The ***Mouse hindbrain*** dataset is available at https://doi.org/10.6084/m9.figshare.24625302.v1 [31].

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interest

The authors declare that they have no competing interest.

Received: 11 November 2024 Accepted: 17 February 2025
Published online: 27 March 2025

References

1. Hedlund E, Deng Q. Single-cell rna sequencing: technical advancements and biological applications. Mol Asp Med. 2018;59:36–46. https://doi.org/10.1016/j.mam.2017.07.003.
2. Ryu Y, Han GH, Jung E, Hwang D. Integration of single-cell RNA-Seq datasets: a review of computational methods. Mol Cells. 2023;46(2):106–19. https://doi.org/10.14348/molcells.2023.0009.
3. Lähnemann D, Köster J, Szczurek E, McCarthy DJ, Hicks SC, Robinson MD, Vallejos CA, Campbell KR, Beerenwinkel N, Mahfouz A, Pinello L, Skums P, Stamatakis A, Attolini CS-O, Aparicio S, Baaijens J, Balvert M, Barbanson Bd, Cappuccio A, Corleone G, Dutilh BE, Florescu M, Guryev V, Holmer R, Jahn K, Lobo TJ, Keizer EM, Khatri I, Kielbasa SM, Korbel JO, Kozlov AM, Kuo T-H, Lelieveldt BPF, Mandoiu II, Marioni JC, Marschall T, Mölder F, Niknejad A, Raczkowska A, Reinders M, Ridder Jd, Saliba A-E, Somarakis A, Stegle O, Theis FJ, Yang H, Zelikovsky A, McHardy AC, Raphael BJ, Shah SP, Schönhuth A. Eleven grand challenges in single-cell data science. Genome Biol. 2020;21(1):31. https://doi.org/10.1186/s13059-020-1926-6.
4. Tran HTN, Ang KS, Chevrier M, Zhang X, Lee NYS, Goh M, Chen J. A benchmark of batch-effect correction methods for single-cell RNA sequencing data. Genome Biol. 2020;21(1):12. https://doi.org/10.1186/s13059-019-1850-9.
5. Chen W, Zhao Y, Chen X, Yang Z, Xu X, Bi Y, Chen V, Li J, Choi H, Ernest B, Tran B, Mehta M, Kumar P, Farmer A, Mir A, Mehra UA, Li J-L, Moos M, Xiao W, Wang C. A multicenter study benchmarking single-cell RNA sequencing technologies using reference samples. Nat Biotechnol. 2021;39(9):1103–14. https://doi.org/10.1038/s41587-020-00748-9.
6. Luecken MD, Theis FJ. Current best practices in single-cell RNA-seq analysis: a tutorial. Mol Syst Biol. 2019;15(6):8746. https://doi.org/10.15252/msb.20188746.

7. Forcato M, Romano O, Bicciato S. Computational methods for the integrative analysis of single-cell data. Brief Bioinf. 2021;22(1):20–9. https://doi.org/10.1093/bib/bbaa042.
8. Haghverdi L, Lun ATL, Morgan MD, Marioni JC. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat Biotechnol. 2018;36(5):421–7. https://doi.org/10.1038/nbt.4091.
9. Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. Nat Biotechnol. 2018;36(5):411–20. https://doi.org/10.1038/nbt.4096.
10. Barkas N, Petukhov V, Nikolaeva D, Lozinsky Y, Demharter S, Khodosevich K, Kharchenko PV. Joint analysis of heterogeneous single-cell RNA-seq dataset collections. Nat Methods. 2019;16(8):695–8. https://doi.org/10.1038/s41592-019-0466-z.
11. Hie B, Bryson B, Berger B. Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nat Biotechnol. 2019;37(6):685–91. https://doi.org/10.1038/s41587-019-0113-3.
12. 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. Nat Methods. 2019;16(12):1289–96. https://doi.org/10.1038/s41592-019-0619-0.
13. Polanski K, Young MD, Miao Z, Meyer KB, Teichmann SA, Park J-E. BBKNN: fast batch alignment of single cell transcriptomes. Bioinformatics. 2019;36(3):964–5. https://doi.org/10.1093/bioinformatics/btz625.
14. Liu J, Gao C, Sodicoff J, Kozareva V, Macosko EZ, Welch JD. Jointly defining cell types from multiple single-cell datasets using LIGER. Nat Protoc. 2020;15(11):3632–62. https://doi.org/10.1038/s41596-020-0391-8.
15. Lotfollahi M, Wolf FA, Theis FJ. scGen predicts single-cell perturbation responses. Nat Methods. 2019;16(8):715–21. https://doi.org/10.1038/s41592-019-0494-8.
16. Amodio M, van Dijk D, Srinivasan K, Chen WS, Mohsen H, Moon KR, Campbell A, Zhao Y, Wang X, Venkataswamy M, Desai A, Ravi V, Kumar P, Montgomery R, Wolf G, Krishnaswamy S. Exploring single-cell data with deep multitasking neural networks. Nat Methods. 2019;16(11):1139–45. https://doi.org/10.1038/s41592-019-0576-7.
17. Lopez R, Regier J, Cole MB, Jordan MI, Yosef N. Deep generative modeling for single-cell transcriptomics. Nat Methods. 2018;15(12):1053–8. https://doi.org/10.1038/s41592-018-0229-2.
18. Xu C, Lopez R, Mehlman E, Regier J, Jordan MI, Yosef N. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. Mol Syst Biol. 2021;17(1):9620. https://doi.org/10.15252/msb.20209620.
19. Gayoso A, Lopez R, Xing G, Boyeau P, Valiollah Pour Amiri V, Hong J, Wu K, Jayasuriya M, Mehlman E, Langevin M, Liu Y, Samaran J, Misrachi G, Nazaret A, Clivio O, Xu C, Ashuach T, Gabitto M, Lotfollahi M, Svensson V, da Veiga Beltrame E, Kleshchevnikov V, Talavera-López C, Pachter L, Theis FJ, Streets A, Jordan MI, Regier J, Yosef N. A Python library for probabilistic analysis of single-cell omics data. Nat Biotechnol. 2022;40(2):163–6. https://doi.org/10.1038/s41587-021-01206-w
20. Virshup I, Bredikhin D, Heumos L, Palla G, Sturm G, Gayoso A, Kats I, Koutrouli M, Berger B, Pe'er D, Regev A, Teichmann SA, Finotello F, Wolf FA, Yosef N, Stegle O, Theis FJ. The scverse project provides a computational ecosystem for single-cell omics data analysis. Nat Biotechnol. 2023;41(5):604–6. https://doi.org/10.1038/s41587-023-01733-8.
21. Luecken MD, Büttner M, Chaichoompu K, Danese A, Interlandi M, Mueller MF, Strobl DC, Zappia L, Dugas M, Colomé-Tatché M, Theis FJ. Benchmarking atlas-level data integration in single-cell genomics. Nat Methods. 2022;19(1):41–50. https://doi.org/10.1038/s41592-021-01336-8.
22. Maan H, Zhang L, Yu C, Geuenich MJ, Campbell KR, Wang B. Characterizing the impacts of dataset imbalance on single-cell data integration. Nat Biotechnol. 2024. https://doi.org/10.1038/s41587-023-02097-9.
23. Zappia L, Phipson B, Oshlack A. Splatter: simulation of single-cell RNA sequencing data. Genome Biol. 2017;18(1):174. https://doi.org/10.1186/s13059-017-1305-0.
24. ...Vieira Braga FA, Kar G, Berg M, Carpaij OA, Polanski K, Simon LM, Brouwer S, Gomes T, Hesse L, Jiang J, Fasouli ES, Efremova M, Vento-Tormo R, Talavera-López C, Jonker MR, Affleck K, Palit S, Strzelecka PM, Firth HV, Mahbubani KT, Cvejic A, Meyer KB, Saeb-Parsy K, Luinge M, Brandsma C-A, Timens W, Angelidis I, Strunz M, Koppelman GH, van Oosterhout AJ, Schiller HB, Theis FJ, van den Berge M, Nawijn MC, Teichmann SA. A cellular census of human lungs identifies novel cell states in health and in asthma. Nat Med. 2019;25(7):1153–63. https://doi.org/10.1038/s41591-019-0468-5.
25. Grün D, Muraro MJ, Boisset J-C, Wiebrands K, Lyubimova A, Dharmadhikari G, van den Born M, van Es J, Jansen E, Clevers H, de Koning EJP, van Oudenaarden A. De novo prediction of stem cell identity using single-cell transcriptome data. Cell Stem Cell. 2016;19(2):266–77. https://doi.org/10.1016/j.stem.2016.05.010.
26. Muraro MJ, Dharmadhikari G, Grün D, Groen N, Dielen T, Jansen E, van Gurp L, Engelse MA, Carlotti F, de Koning EJP, van Oudenaarden A. A single-cell transcriptome atlas of the human pancreas. Cell Syst. 2016;3(4):385–3943. https://doi.org/10.1016/j.cels.2016.09.002.
27. Lawlor N, George J, Bolisetty M, Kursawe R, Sun L, Sivakamasundari V, Kycia I, Robson P, Stitzel ML. Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes. Genome Res. 2017;27(2):208–22. https://doi.org/10.1101/gr.212720.116.
28. Baron M, Veres A, Wolock SL, Faust AL, Gaujoux R, Vetere A, Ryu JH, Wagner BK, Shen-Orr SS, Klein AM, Melton DA, Yanai I. A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. Cell Syst. 2016;3(4):346–3604. https://doi.org/10.1016/j.cels.2016.08.011.
29. Xin Y, Kim J, Okamoto H, Ni M, Wei Y, Adler C, Murphy AJ, Yancopoulos GD, Lin C, Gromada J. RNA sequencing of single human islet cells reveals type 2 diabetes genes. Cell Metab. 2016;24(4):608–15. https://doi.org/10.1016/j.cmet.2016.08.018.
30. Segerstolpe A, Palasantza A, Eliasson P, Andersson E-M, Andréasson A-C, Sun X, Picelli S, Sabirsh A, Clausen M, Bjursell MK, Smith DM, Kasper M, Ammala C, Sandberg R. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. Cell Metab. 2016;24(4):593–607. https://doi.org/10.1016/j.cmet.2016.08.020.
31. Vladoiu MC, El-Hamamy I, Donovan LK, Farooq H, Holgado BL, Sundaravadanam Y, Ramaswamy V, Hendrikse LD, Kumar S, Mack SC, Lee JJY, Fong V, Juraschka K, Przelicki D, Michealraj A, Skowron P, Luu B, Suzuki H, Morrissy AS, Cavalli FMG, Garzia L, Daniels C, Wu X, Qazi MA, Singh SK, Chan JA, Marra MA, Malkin D, Dirks P, Heisler L, Pugh T, Ng K,

Notta F, Thompson EM, Kleinman CL, Joyner AL, Jabado N, Stein L, Taylor MD. Childhood cerebellar tumours mirror conserved fetal transcriptional programs. Nature. 2019;572(7767):67–73. https://doi.org/10.1038/s41586-019-1158-7.
32. Cusanovich DA, Hill AJ, Aghamirzaie D, Daza RM, Pliner HA, Berletch JB, Filippova GN, Huang X, Christiansen L, DeWitt WS, Lee C, Regalado SG, Read DF, Steemers FJ, Disteche CM, Trapnell C, Shendure J. A single-cell atlas of in vivo mammalian chromatin accessibility. Cell. 2018;174(5):1309–132418. https://doi.org/10.1016/j.cell.2018.06.052.
33. 10x Genomics: Fresh cortex from adult mouse brain (P50) 2019. https://support.10xgenomics.com/single-cell-atac/datasets/1.2.0/atac_v1_adult_brain_fresh_5k? Accessed 2024-10-22
34. Fang R, Preissl S, Li Y, Hou X, Lucero J, Wang X, Motamedi A, Shiau AK, Zhou X, Xie F, Mukamel EA, Zhang K, Zhang Y, Behrens MM, Ecker JR, Ren B. Comprehensive analysis of single cell ATAC-seq data with SnapATAC. Nat Commun. 2021;12(1):1337. https://doi.org/10.1038/s41467-021-21583-9.
35. Wolf FA, Angerer P, Theis FJ. SCANPY: large-scale single-cell gene expression data analysis. Genome Biol. 2018;19(1):15. https://doi.org/10.1186/s13059-017-1382-0.
36. Yu X, Xu X, Zhang J, Li X. Batch alignment of single-cell transcriptomics data using deep metric learning. Nat Commun. 2023;14(1):960. https://doi.org/10.1038/s41467-023-36635-5.
37. Single-cell RNA-seq analysis workshop. Teaching materials at the Harvard Chan Bioinformatics Core 2023. https://github.com/hbctraining/scRNA-seq Accessed 2023-10-14
38. Cristian P-M, Aarón V-J, Armando E-HD, Estrella MLY, Daniel N-R, Paul S-CJ, David G-V, Osbaldo R-A. Diffusion on PCA-UMAP manifold captures a well-balance of local, global, and continuum structure to denoise single-cell RNA sequencing data. bioRxiv 2022. https://doi.org/10.1101/2022.06.09.495525
39. Stuart T, Butler A, Hoffman P, Hafemeister C, Papalexi E, Mauck WM, Hao Y, Stoeckius M, Smibert P, Satija R. Comprehensive integration of single-cell data. Cell. 2019;177(7):1888–190221. https://doi.org/10.1016/j.cell.2019.05.031.
40. Rubner Y, Tomasi C, Guibas LJ. A metric for distributions with applications to image databases. In: Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), 1998;59–66. https://doi.org/10.1109/ICCV.1998.710701
41. ...Flamary R, Courty N, Gramfort A, Alaya MZ, Boisbunon A, Chambon S, Chapel L, Corenflos A, Fatras K, Fournier N, Gautheron L, Gayraud NTH, Janati H, Rakotomamonjy A, Redko I, Rolet A, Schutz A, Seguy V, Sutherland DJ, Tavenard R, Tong A, Vayer T. POT: python optimal transport. J Mach Learn Res. 2021;22(78):1–8.
42. 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. Nat Methods. 2019;16(12):1289–96. https://doi.org/10.1038/s41592-019-0619-0.
43. Xiong L, Tian K, Li Y, Ning W, Gao X, Zhang QC. Online single-cell data integration through projecting heterogeneous datasets into a common cell-embedding space. Nat Commun. 2022;13(1):6118. https://doi.org/10.1038/s41467-022-33758-z.
44. Büttner M, Miao Z, Wolf FA, Teichmann SA, Theis FJ. A test metric for assessing single-cell RNA-seq batch correction. Nat Methods. 2019;16(1):43–9. https://doi.org/10.1038/s41592-018-0254-1.
45. Mandric I, Hill BL, Freund MK, Thompson M, Halperin E. BATMAN: fast and accurate integration of single-cell RNA-seq datasets via minimum-weight matching. iScience. 2020;23(6): 101185. https://doi.org/10.1016/j.isci.2020.101185.

Publisher's Note