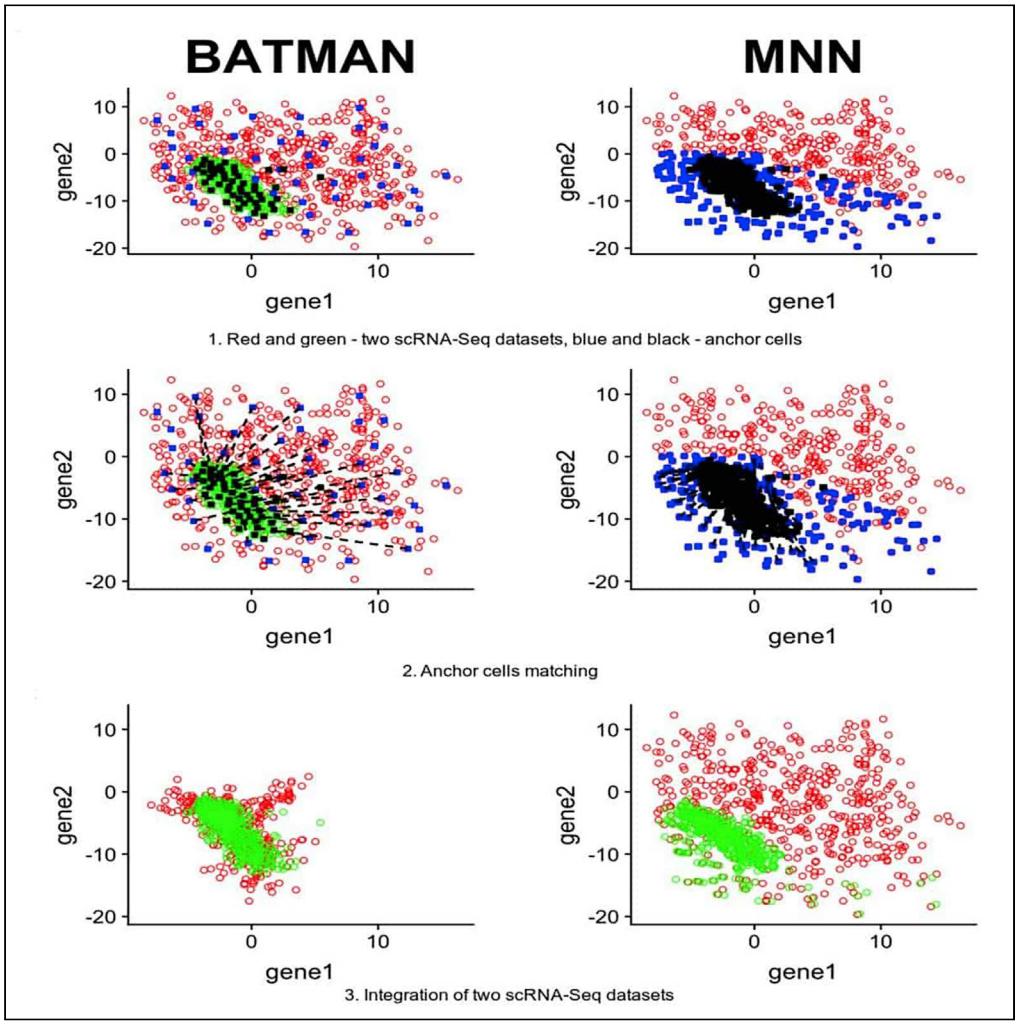
iScience

CellPress
OPEN ACCESS

Article

BATMAN: Fast and Accurate Integration of Single-Cell RNA-Seq Datasets via Minimum-Weight Matching



Igor Mandric, Brian L. Hill, Malika K. Freund, Michael Thompson, Eran Halperin

imandric@ucla.edu (I.M.)
ehalperin@cs.ucla.edu (E.H.)

HIGHLIGHTS

Current methods for scRNA-seq dataset integration are based on MNN paradigm

MNN paradigm has drawbacks, e.g., it fails in case of non-orthogonal batch effects

BATMAN proposes a new paradigm based on minimum-weight bipartite matching

BATMAN outperforms the existing scRNA-seq integration methods in the gene space

Mandric et al., iScience *23*, 101185
June 26, 2020
https://doi.org/10.1016/j.isci.2020.101185

iScience

CellPress
OPEN ACCESS

Article

BATMAN: Fast and Accurate Integration of Single-Cell RNA-Seq Datasets via Minimum-Weight Matching

Igor Mandric,[1,6,*] Brian L. Hill,[1] Malika K. Freund,[2] Michael Thompson,[1] and Eran Halperin[1,2,3,4,5,*]

SUMMARY

Single-cell RNA-sequencing (scRNA-seq) is a set of technologies used to profile gene expression at the level of individual cells. Although the throughput of scRNA-seq experiments is steadily growing in terms of the number of cells, large datasets are not yet commonly generated owing to prohibitively high costs. Integrating multiple datasets into one can improve power in scRNA-seq experiments, and efficient integration is very important for downstream analyses such as identifying cell-type-specific eQTLs. State-of-the-art scRNA-seq integration methods are based on the mutual nearest neighbor paradigm and fail to both correct for batch effects and maintain the local structure of the datasets. In this paper, we propose a novel scRNA-seq dataset integration method called BATMAN (BATch integration via minimum-weight MAtchiNg). Across multiple simulations and real datasets, we show that our method significantly outperforms state-of-the-art tools with respect to existing metrics for batch effects by up to 80% while retaining cell-to-cell relationships.

INTRODUCTION

Single-cell RNA sequencing (scRNA-seq) has revolutionized transcriptomics as it enables the computational inference of cell types, the discovery of new cell states, and the reconstruction of cellular differentiation trajectories (Angerer et al., 2017). Although the scale of the datasets produced by scRNA-seq is continuously growing (Svensson et al., 2018), the demand for sequencing an even larger number of cells greatly exceeds the current throughput of sequencing experiments (Brennecke et al., 2013) and, therefore, cells must be processed in multiple sequencing runs, or batches. Owing to the high level of technical noise, systematic differences between sequencing instruments, and other confounding factors, simple concatenation is a suboptimal approach to integrate multiple batches of a dataset. Furthermore, batch effects have been shown to cause an increased number of false positives in downstream analyses (Goh et al., 2017). To mitigate the level of false discoveries, a proper integration of multiple batches must eliminate the differences caused by batch effects.

Current methods for merging scRNA-seq datasets can conventionally be divided into two categories: batch correction and integration methods (Hie et al., 2019). Batch correction is the adjustment of gene expression in the high-dimensional gene space to account for confounding variation between technical scRNA-seq replicates—in other words, batch correction operates on the gene expression levels themselves. Integration methods operate instead in a latent low-dimensional space, such as canonical correlation analysis (CCA) embeddings or embeddings learned from neural networks (Lopez et al., 2018), and are applied to the problem of merging multiple datasets across different technologies or biological conditions. Batch correction methods are more interpretable since they allow for a wider range of downstream analyses including differential gene expression and pseudo-time trajectory inference. On the other hand, integration methods enjoy a limited spectrum of applications, the most frequently used being visualization and cell-type classification. Throughout this paper, we will use the term "integration" for combining scRNA-seq datasets into one as it is more general.

In this paper, we present a method for integration of single-cell datasets called BATMAN (BATch integration via minimum weight MAtchiNg), based on a parsimonious one-to-one matching of representative cells across datasets that maintains local structure. BATMAN operates in the high-dimensional gene space and

[1]Department of Computer Science, University of California Los Angeles, Los Angeles, CA, USA

[2]Department of Human Genetics, David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA, USA

[3]Department of Anesthesiology and Perioperative Medicine, David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA, USA

[4]Department of Computational Medicine, David Geffen School of Medicine, University of California Los Angeles, Los Angeles, CA, USA

[5]Institute of Precision Health, University of California, Los Angeles, CA, USA

[6]Lead Contact

*Correspondence: imandric@ucla.edu (I.M.), ehalperin@cs.ucla.edu (E.H.)

https://doi.org/10.1016/j.isci.2020.101185

uses the minimal total correction necessary to remove discrepancies between datasets. We show that BATMAN not only significantly outperforms state-of-the-art tools in terms of widely used integration quality metrics on a wide range of simulated and real datasets (by 80%) but also maintains the local structure of each dataset.

Background

Given two scRNA-seq datasets, a query dataset D_1 and a reference dataset D_2, the goal of integration is to align the query to the reference dataset by removing the confounding variation between them (Stuart et al., 2019). Usually after alignment only the query dataset is modified, i.e., the gene expression values of the cells in D_1 are modified, whereas the cells in D_2 remain intact. The quality of the alignment is generally measured by a metric of how well mixed the two cell populations are after modification. D_1 and D_2 are considered well mixed if the local dataset-label distribution in the neighborhood of each cell (in the context of the integrated dataset D_3) matches the global dataset label distribution, i.e., every ball containing cells of D_3 contains cells from D_1 and D_2 in the same proportion as given by their cell counts (Büttner et al., 2019). The alignment is constrained by the local structure of each dataset; after integration, the neighborhood relationship among the cells in the query dataset must be preserved. If the datasets consist of several cell types, then the above definition applies to each cell type separately, i.e., cells of cell type x of dataset D_1 should only be mixed with cells of cell type x of dataset D_2.

scRNA-seq dataset integration by maximizing the mixing quality of two datasets under the constraint of local structure preservation is a challenging task as there are no straightforward or trivial objective functions to optimize over. Current state-of-the-art scRNA-seq integration metrics such as kBET (Büttner et al., 2019) (k-nearest neighbor batch effect test) measure the quality of mixing through the concordance of the global and local dataset label distributions among the nearest k neighbors. Using such metrics as objectives in the integration problem prevents any standard optimization techniques from being applicable. Although it is possible to find a perfect mixing with respect to the aforementioned metrics by random assignment of cells in the dataset D_1 to the cells of dataset D_2, the biological signal of each dataset would be compromised as the local structure of each dataset would be destroyed.

One approach for solving the integration problem for scRNA-seq datasets is to use tools designed for bulk RNA-seq data. In this case, each cell in an scRNA-seq dataset is viewed as a single bulk RNA-seq sample. Nonetheless, integration methods that are borrowed from bulk RNA-seq analysis depend on normalization techniques and usually assume that the data comes from a particular distribution. For example, ComBat (Johnson et al., 2007) and limma (Ritchie et al., 2015) assume that the observed expression values come from a Gaussian distribution. These assumptions may not hold true for some real datasets (Pierson and Yau, 2015; Risso et al., 2018). Additionally, ComBat and limma were not designed to handle datasets consisting of multiple cell types (Wang et al., 2019). As it has been previously shown, in more complicated scenarios characteristic to single-cell datasets, these methods perform poorly (Wang et al., 2019).

Many methods have recently been proposed for integration of scRNA-seq data, primarily based on clustering or deep neural networks, for example, BERMUDA (Wang et al., 2019), SAUCIE (Amodio et al., 2019), and Harmony (Korsunsky et al., 2019). BERMUDA uses a deep neural network architecture called an autoencoder to learn a low-dimensional representation of the data in an unsupervised manner. The autoencoder network is trained to minimize the loss between the original gene expression values and the reconstructed values after performing the dimensionality reduction, as well as the maximum mean discrepancy (MMD) between similar clusters from different batches. Like BERMUDA, SAUCIE also uses a sparse autoencoder to learn a low-dimensional representation of the data. The approach adds several novel regularization methods to the network activations to encourage the network to learn representations which are useful for several tasks in single-cell analyses such as clustering and integration. Harmony is an integration method that uses soft clustering after projecting the cells into a low-dimensional space by principal components analysis (PCA) to iteratively find cluster centroids, which are then used to calculate a cell-specific correction factor. The downside of these methods is that they operate in latent space, which limits their interpretability and use in downstream analyses such as differential gene expression and single-cell eQTL analyses.

Another group of integration methods that operate in gene space includes Seurat v3.0 (Stuart et al., 2019), MNNcorrect (Haghverdi et al., 2018), and Scanorama (Hie et al., 2019). MNNcorrect finds similar pairs of

cells across batches where both cells are contained in each other's set of nearest neighbors (mutual nearest neighbors, or MNN). The average difference in the gene expression between many pairs of mutual nearest neighbors estimates the batch effect, and this estimate can be used to correct the expression values. Seurat (version 3.0) builds on the MNN methodology, using MNN to determine "anchor points." For dimensionality reduction, Seurat uses canonical correlation analysis (CCA) to find a subspace common to all datasets, which should be void of technical variation that is local to each dataset (Stuart et al., 2019; Thompson et al., 2019). Scanorama also uses MNN for integration and batch correction, but the MNN search is performed in a low-dimensional space after randomized singular value decomposition (SVD) and uses a faster approximate nearest neighbor search to improve scalability. Additionally, Scanorama was designed to handle the alignment of multiple datasets without being sensitive to the ordering of alignment.

RESULTS

We compared BATMAN with three other state-of-the-art tools that operate in the gene space: Seurat v3.0, MNNcorrect, and Scanorama. We did not include tools such as limma and ComBat in the comparison, as they are more appropriate for bulk RNA-seq analysis and have been shown to fail in more complicated scenarios characteristic of single-cell datasets (Wang et al., 2019). We also compared BATMAN with Harmony, which is the state-of-the-art integration tool in PC space.

Evaluation Metrics

Traditionally, scRNA-seq dataset integration quality has been assessed visually using UMAP and/or tSNE plots. However, there are multiple quantitative evaluation metrics available (Haghverdi et al., 2018; Stuart et al., 2019; Wang et al., 2019). All of the metrics are based on scanning local neighborhoods of the cells in the combined dataset (i.e., after integration) and testing if the proportion of cells from the two datasets is the same as globally, for example, using the entropy mixing score. A novel test, kBET (Büttner et al., 2019), was designed independently of all the batch correction and integration methods and tests whether the local dataset label distribution is concordant with the global. As noted in (Büttner et al., 2019), kBET has an issue: it fails to measure batch effects properly when the datasets have different cell-type compositions. Additionally, the more standard metrics such as entropy mixing score lack interpretability. To overcome these issues, we use LISI (Local Inverse Simpson Index), a novel recently proposed metric (Korsunsky et al., 2019). To compute LISI score, one has to build Gaussian kernel-based distributions of neighborhoods. Then, for each neighborhood the Inverse Simpson Index is computed:

$$S = \frac{1}{\sum_{b=1}^{B} p(b)},$$

where $p(b)$ refers to the probability of batch b in the local neighborhood. LISI score is then reported as the average value of S across all neighborhoods. Its value ranges from 1 to 2, and it has a simple interpretation as the expected number of cells needed to be sampled before two are drawn from the same dataset. In our evaluations, we use two versions of LISI: integration LISI (iLISI; 1 is perfect separability, 2 is perfect mixing) and cell-type LISI (cLISI; 1, all cell types are separable; 2, cell types are mixed with each other).

To quantify how well a method preserves the local structure of datasets after integration, we measure the average percentage of retained nearest neighbors. Intuitively, if a cell and its neighbors are not perturbed by a batch correction method, then they will have the same set of nearest neighbors before and after applying the correction. This quantity is computed as follows: given two datasets D_1 and D_2, for each cell we determine the top-k nearest neighbors in the original dataset. Next, for each cell we determine the top-k nearest neighbors after integration (again, in the context of its original dataset). We report the average percentage of retained nearest neighbors across all the cells in the union of D_1 and D_2. We will refer to this metric as k-RNN (retained nearest neighbors).

Integration of Simulated Datasets

We simulated scRNA-seq datasets based on a gamma-Poisson distribution using the *splatter* (Zappia et al., 2017) simulator (version 1.10.0) to compare BATMAN with Seurat V3.0, MNNcorrect (from scran R package, version 1.12.1), and Scanorama (version 1.5) across five different scenarios:

1. Large batch effects (LB)
2. Large batch effects with large dropout rate (LB-DR)

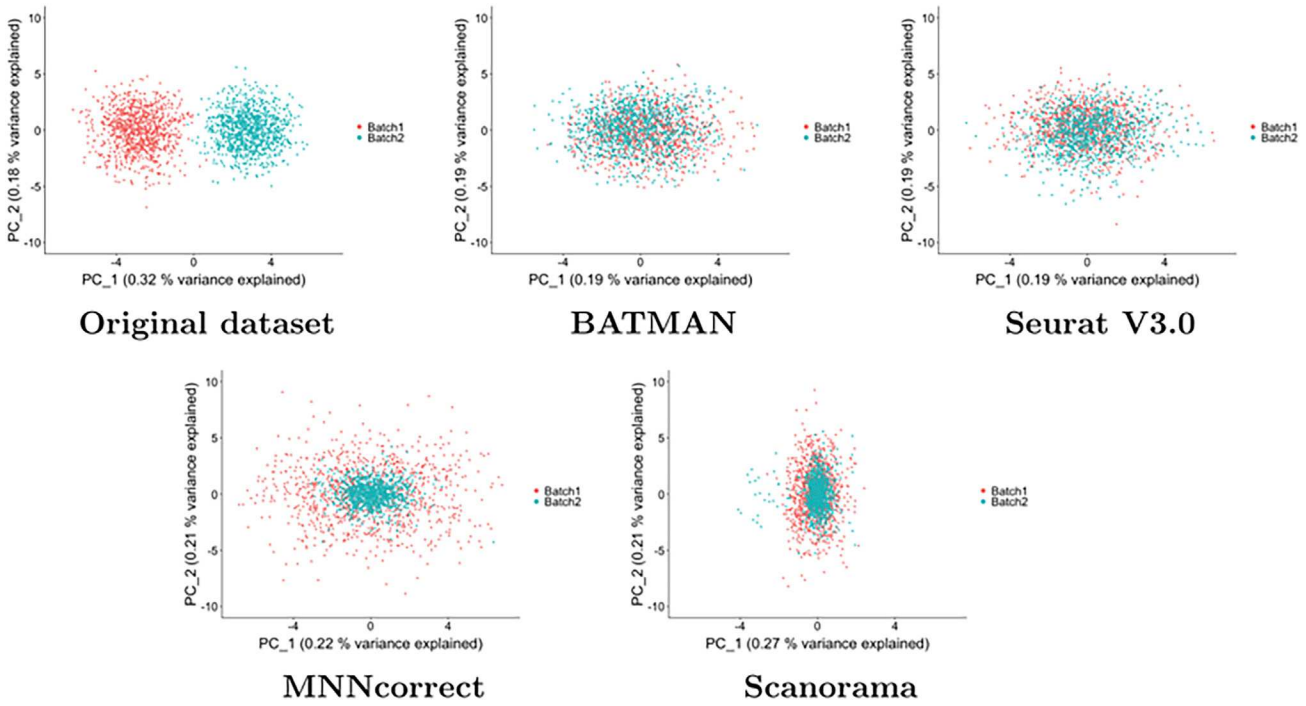


Figure 1. Simulated Datasets with Large Batch Effects (LB Scenario)—PCA Plots
Each dataset consists of 1,000 cells and 1,000 genes. The top two PCs are plotted.

3. Large batch effects with unequal batch sizes (LB-UB)
4. Small batch effects (SB)
5. Large batch effects—two cell types (LB-CT)

Within Splatter, the magnitude of batch effects is controlled with two parameters: *batch.facLoc* and *batch.facScale*. Per Splatter's documentation, for large batch effects scenarios in simulations we set both parameters to 0.5 and for small batch effects we set both to 0.001. The *dropout.shape* parameter controls the magnitude of dropout rate; setting this parameter to larger values produces sparser scRNA-seq datasets. This parameter was set to 2 for the LB-DR scenario and set to 1 in all other scenarios. In all five scenarios, we simulated 1,000 genes. Batches in the LB-UB scenario consist of 200 and 1,000 cells, whereas in the other scenarios each batch has 1,000 cells. Finally, in the LB-CT scenario, each dataset consists of two cell types (80% and 20% cell type frequency in both batches).

For each scenario, we simulated 100 datasets and ran BATMAN alongside the other three tools. We computed the average values of iLISI and 50-RNN across the 100 runs and along with confidence intervals (CI) for each metric. The lower bound of the CIs were computed as the average of the second and the third value in the sorted list of 100 values, and the upper bound of the CIs were computed as the average of the 97th and the 98th values correspondingly. All the tools were run with their default parameters. In all the experiments, we used Seurat V3.0 pre-processing steps to obtain log-counts for each dataset.

In the LB scenario, a visual inspection of the datasets in the space of the top 2 principal components (see Figure 1) suggests that Seurat V3.0 slightly undercorrected for the batch effects. MNNcorrect and Scanorama, although correctly shifting the datasets toward the centers of each other, failed to properly account for the variance of the datasets as evidenced by Figure 1. However, BATMAN is the only tool that properly integrated two datasets; we show that the iLISI score of the integrated dataset is higher after integration only when applying BATMAN and the other tools failed to properly correct for the differences between the datasets as evidenced by very low iLISI scores (see Table 1). For example, although the PCA plot of Seurat V3.0 shows that the batch effects were mostly removed (Figure 1), its low iLISI score suggests that, in fact, it failed to properly integrate the two datasets. To investigate this discrepancy between the iLISI results and the PCA plots, for each integration result, we

Metric	Original	BATMAN	Seurat V3.0	MNNcorrect	Scanorama
Mean iLISI	1.82	**1.84**	1.00	1.00	1.01
CI iLISI	(1.57, 1.92)	**(1.51, 1.96)**	(1.00, 1.02)	(1.00, 1.01)	(1.00, 1.01)
Mean 50-RNN	1.00	**0.92**	0.58	0.51	0.14
CI 50-RNN	(1.00, 1.00)	**(0.90, 0.95)**	(0.56, 0.61)	(0.51, 0.52)	(0.11, 0.16)

Table 1. Integration Results in LB Scenario (Large Batch Effects): iLISI and 50-RNNscores
The best results are emphasized in bold. CI stands for confidence interval.

computed iLISI scores based on different numbers of top principal components (Figure 2A). When we consider top 2 principal components, the iLISI scores of Seurat V3.0 are high. However, as we increase the dimensionality of the dataset by projecting it to a larger number of principal components, the integration quality of Seurat V3.0 is constantly decreasing and in the limit it tends to 1. On the contrary, the original datasets look linearly separated in the top 2 principal components and the iLISI score is 1. However, when considering more top principal components, Figure 2A shows that the two original datasets become better mixed with each other. BATMAN has the highest performance not only when considering the top two principal components but also when considering more top principal components. It is the only method that manages to efficiently maintain a high iLISI score in a larger number of dimensions.

Notably, BATMAN is the only tool that not only properly corrected for the batch effects but also preserved the biological signal characteristic to each dataset before integration (see Table 1). The 50-RNN of BATMAN equal to 0.92 means that, on average, BATMAN preserves 92% of each cell's 50 nearest neighbors after integration. Seurat V3.0 and MNN each preserves 58% and 51% correspondingly, whereas Scanorama integrates two datasets without keeping their local structure resulting in a poor 50-RNN score of 0.14. BATMAN, Seurat V3.0, and Scanorama display a consistent *k*-RNN score across multiple values of *k*, whereas Scanorama has very poor results for smaller values of *k* (reflecting that it destroys local structure in each small neighborhood of the two datasets; Figure 2B).

Higher dropout rates (the LB-DR scenario) do not seem to drastically change the results of integration across the four methods (Figure S5) as compared with the LB scenario. Again, BATMAN significantly outperformed the other tools on iLISI and 50-RNN scores (see Table S1).

When the two datasets consist of different numbers of cells (the LB-UB scenario), BATMAN is the only method that demonstrates any improvement in the iLISI score after integration (see Figure S6 and Table S2). Notably, both BATMAN and Seurat V3.0 preserved the local structure of the datasets with a 50-RNN score above 0.7 (see Figure S6 and Table S2).

We also measured the ability of the four methods to properly correct for very small batch effects (the SB scenario, see Figure S7, Supplemental Information). In this case, the two datasets were well mixed even before integration with an initial iLISI score of 1.88. Given that any integration method will attempt to correct for batch effects even when the datasets are perfectly mixed, in this scenario a decreased iLISI score after integration indicates an overcorrection. As expected, the iLISI score for each of the four methods is smaller than the score before integration (see Table S3). However, the smallest overcorrection is achieved by BATMAN (iLISI = 1.87), whereas the other methods' overcorrection introduced large batch effects.

Finally, when two cell types are simulated (the LB-CT scenario), MNNcorrect, BATMAN, and Scanorama achieve high iLISI scores (of at least 1.6), whereas Seurat V3.0 fails to properly integrate the two datasets. Interestingly, MNNcorrect slightly outperformed BATMAN both in terms of iLISI and 50-RNN scores (see Figure S8 and Table S4).

The analysis of iLISI and k-RNN scores in PCA embeddings suggest that, in most of the simulation scenarios, MNN-based methods only work well in latent spaces of up to 100 principal components, whereas BATMAN successfully maintains a high quality of integration in higher dimensions. BATMAN also retains the local structure by preserving cell-to-cell relationships for each integrated dataset, whereas the MNN-based methods perform worse (see Figures S9–S12).

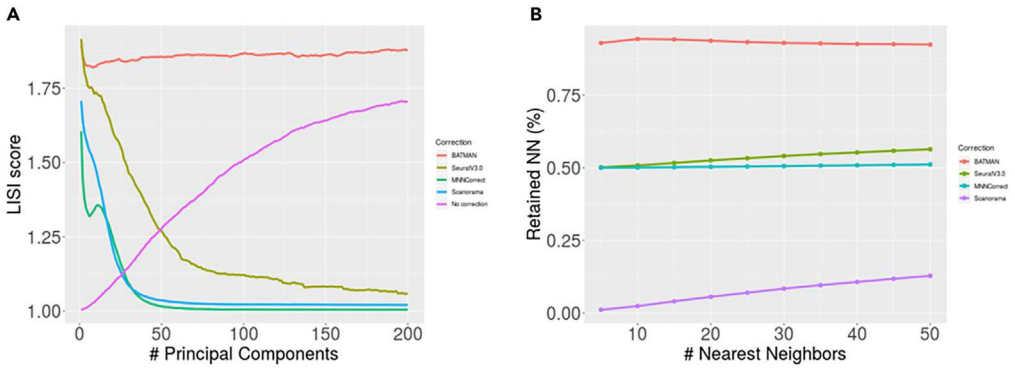


Figure 2. Evaluation of Integration with Large Batch Effects (LB Scenario)

(A) iLISI score as a function of the number of top principal components. (B) *k*-RNN score for different values of *k*.

Integration of Real Datasets across Different Technologies

To compare BATMAN with the other tools in the case of real datasets across different single-cell technologies, we downloaded two pancreatic datasets: a CEL-Seq2 dataset consisting of 2,285 cells (Muraro et al., 2016) and a Smart-Seq2 dataset consisting of 2,394 cells (Segerstolpe et al., 2016). Both datasets include 13 pancreatic cell types with different cell-type composition (see Table S5 for cell-type composition and Figure S13 for UMAP plots).

To examine how different methods correct for batch effects and match cell-type populations, we performed two experiments: (1) integration of full datasets ("all cell types") and (2) integration with one cell type missing from one of the datasets ("1-held-out"). The second experiment evaluates whether the integration methods can align cells of the same cell types across datasets. For maximum stringency, we excluded the most abundant cell type (alpha cells, 37% of cells in CEL-Seq2 dataset and 42% of cells in Smart-Seq2 dataset—see Table S5) from CEL-Seq2 dataset.

First, when all cell types are present across the two datasets, all four methods perform well on matching cell-type populations; the cell types are initially well separable (cLISI score 1.05, Table 2) and they remain so after the integration of all four methods. However, there are large batch effects between the two datasets (iLISI score 1.07). Despite the fact that the integration results look successful upon visual inspection for all methods (see UMAP plots of Figures 3 and S14 of Supplemental Information), BATMAN outperformed the MNN-based methods (iLISI of 1.55 for BATMAN versus next highest iLISI of 1.35 for Seurat V3.0).

Notably, BATMAN is outperformed by Seurat V3.0 and MNNcorrect in lower dimensions as iLISI scores for BATMAN are slightly worse than those for Seurat V3.0 and MNNcorrect in top-50 to top-170 principal components (Figure 4A). However, as the number of dimensions grows, BATMAN's iLISI score increases and the iLISI scores of MNN-based methods decrease. BATMAN showed significantly higher *k*-RNN scores compared with the other methods (Figure 4B), which demonstrates that BATMAN better preserves the local structure of each dataset.

Experiment	Score	No Correction	BATMAN	Seurat V3.0	MNNcorrect	Scanorama
All cell types	iLISI	1.07	**1.55**	1.35	1.18	1.22
	cLISI	1.05	1.08	**1.03**	1.08	1.04
1-Held-out	iLISI	1.09	**1.39**	1.23	1.20	1.21
	cLISI	1.04	1.07	**1.03**	1.09	1.05

Table 2. Integration Results for Real Pancreas Dataset

The best results (of four methods) are marked in bold. For iLISI, higher corresponds to higher quality of integration; for cLISI, lower scores correspond to better separation of cell types.

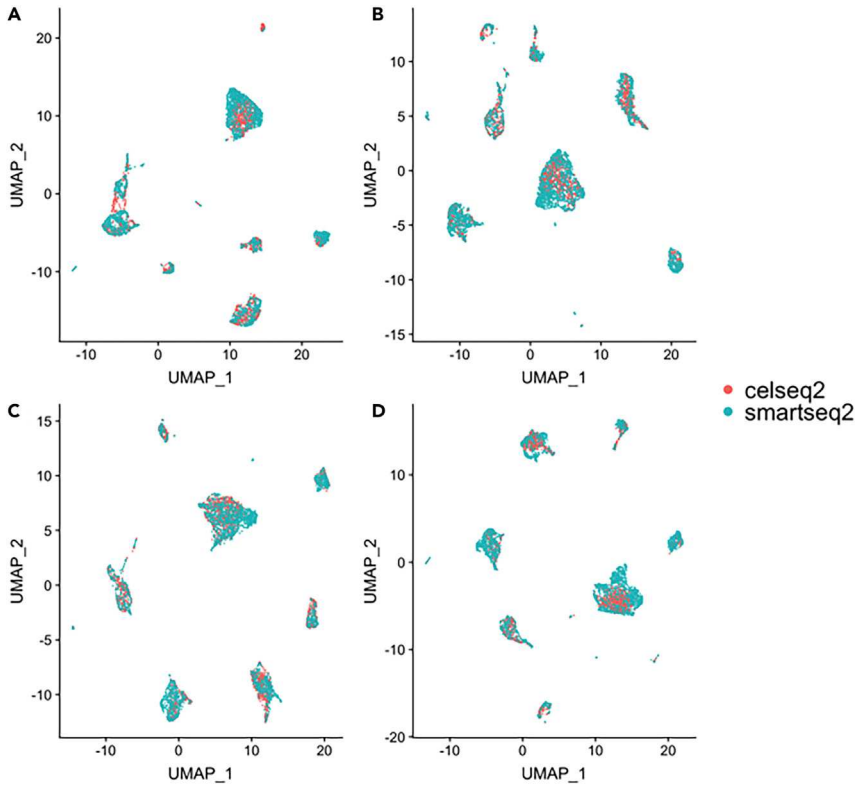


Figure 3. Integration of the Two Pancreatic Datasets across Different Technologies—UMAP Plots ("All Cell Types" Scenario)

(A) BATMAN; (B) Seurat V3.0; (C) MNNcorrect; (D) Scanorama.

Second, in the 1-held-out experiment, all methods correctly matched the cell types leaving alpha cells from Smart-Seq2 dataset unmatched (Table 2 and Figures S15 and S16). The cLISI scores of all the methods are approximately the same, whereas the iLISI scores vary. BATMAN achieves the best iLISI score (1.39), whereas the worst iLISI scores are produced by MNNcorrect and Scanorama integration results. Again, BATMAN introduced the least deformation to the local structure of the datasets compared with the MNN-based methods (Figure S17).

In terms of runtime, BATMAN and Scanorama perform best. In the "all cell types" experiment, BATMAN and Scanorama integrated the two datasets in 9 and 14 s, whereas Seurat V3.0 and MNNcorrect, in 22.3

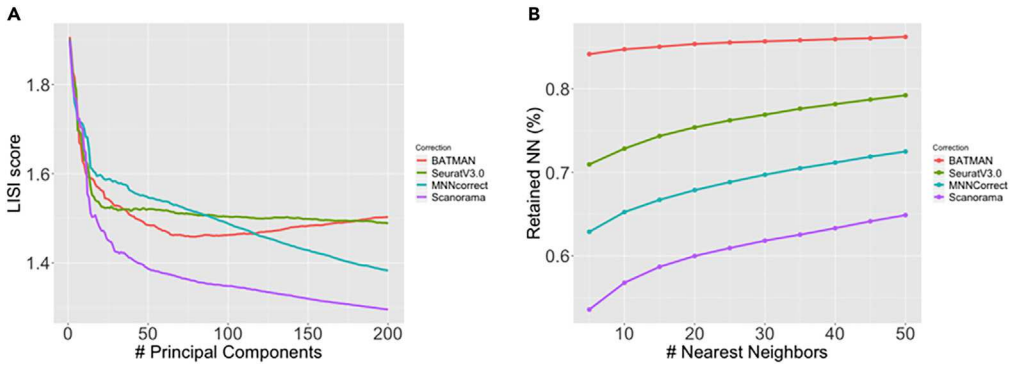


Figure 4. Integration of the Two Pancreatic Datasets across Different Technologies: Integration Quality and Local Structure Preserving

(A) iLISI score as a function of top principal components; (B) *k*-RNN metric across different values of *k*.

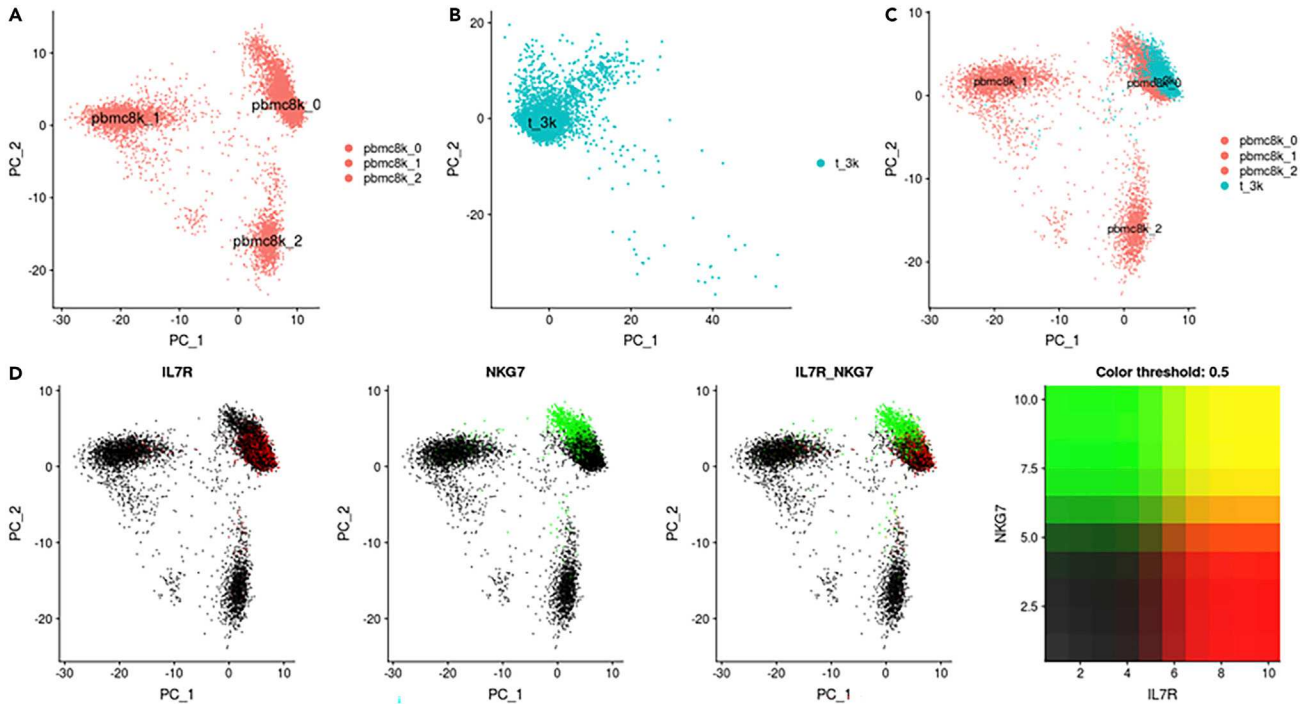


Figure 5. PBMC 8k and Pan T Cells Datasets (10X Genomics)

(A) PCA plot of PBMC 8k dataset reveals three clusters: pbmc8k_0, pbmc8k_1, and pbmc8k_2. (B) PCA plot of Pan T cells dataset reveals one cluster—t_3k. (C) PCA plot of both datasets with no integration shows significant batch effects between cluster pbmc8k_0 of PBMC 8k dataset and t_3k cluster of Pan T cells dataset. (D) Feature plot showing co-expression of IL7R and NKG7 marker genes. Pan T cells dataset corresponds to cluster pbmc8k_0 of PBMC 8k dataset. A proper integration of these two datasets should mix these two clusters.

and 228.6 s correspondingly. In the 1-held-out experiment, the runtimes were slightly smaller: BATMAN, 4 s; Scanorama, 11.8 s; Seurat V3.0, –16.3 s; and MNNcorrect, 190.8 s. All the experiments were performed on a MacBook Pro laptop (16 GB RAM, 8 core 2.3 GHz Intel Core i9 processor).

Integration of Real 10X Genomics Datasets

Current trends in scRNA-seq suggest that droplet-based technologies such as 10X Genomics will become the state of the art in single-cell data acquisition owing to their ability to simultaneously profile large numbers of cells in one experiment. Also, their cost-effectiveness makes them more attractive for large-scale experiments. Therefore, it is crucial for integration methods to be able to handle massive and sparse datasets like those of 10X Genomics, which have a higher dropout rate.

We sought to assess the ability of BATMAN to integrate two 10X Genomics datasets with different numbers of cells and different cell-type composition. For this purpose, we downloaded the PBMC 8k dataset (8,381 peripheral blood mononuclear cells) and the ''Pan T cells'' dataset (3,555 T cells) from 10X Genomics' official website. The cells in the PBMC 8k dataset belong to multiple cell types such as CD4 T cells, B cells, dendritic cells, and more; thus, we expect the Pan T cells dataset to be biologically similar to a subset of the PBMC 8k dataset. Indeed, the PCA plots reveal that the PBMC 8k dataset consists of three clusters (Figure 5A; ''pbmc8k_0,'' ''pbmc8k_1,'' and ''pbmc8k_2''), whereas the Pan T cells dataset consists of a single cluster (Figure 5B; ''t_3k''). On a PCA plot of the combined dataset without integration in Figure 5C, t_3k is closer to pbmc8k_0. The expression plots of two immune marker genes IL7R and NKG7 (Figure 5D) reveal a shared biological signal between the two clusters. However, large batch effects are present (Figure 5C).

We integrated the two datasets using BATMAN and the three MNN-based methods. BATMAN not only had the highest iLISI score of the four methods but also better preserved cell-to-cell relationships (Figure 6A). MNNcorrect performed slightly worse than BATMAN in terms of both integration and local

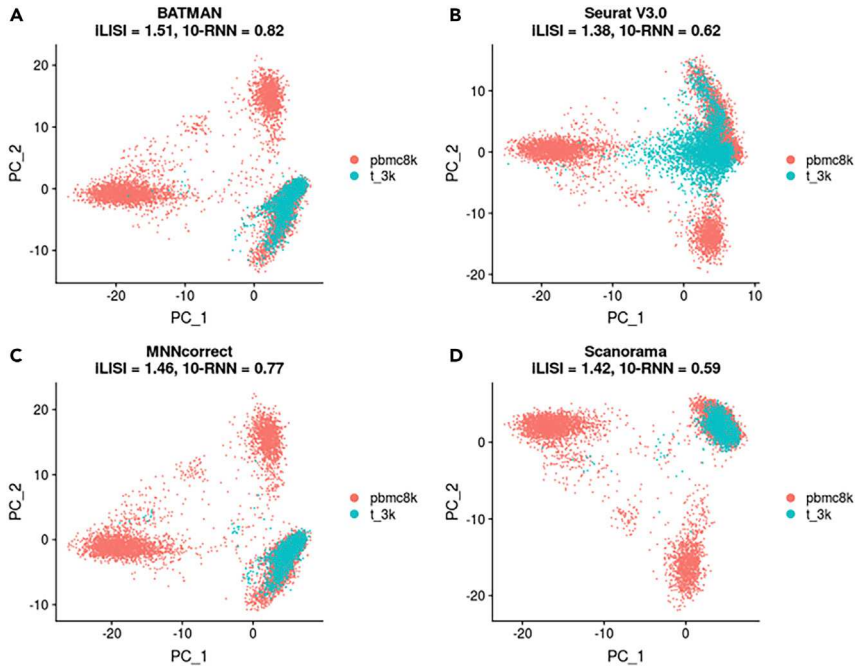


Figure 6. Integration of PBMC 8k and Pan T Cells Datasets (10X Genomics)
(A) BATMAN; (B) Seurat V3.0; (C) MNNcorrect; (D) Scanorama.

structure preserving (Figure 6C). Although Scanorama had a good iLISI score, it destroyed much of the cell-to-cell relationships (Figure 6D). Finally, Seurat V3.0 failed to properly match the biological state of the cells between the two datasets (Figure 6B), which resulted in lower quality of integration.

In terms of runtime, BATMAN and Scanorama were the fastest tools finishing the integration task of the two 10X Genomics datasets in 20.1 and 31.0 s correspondingly. Seurat V3.0 used 211.2 s, whereas MNNcorrect was the slowest tool—1,099.5 s.

Integration in PCA Space

We compared BATMAN against Harmony (version 0.1) on integrating the two PBMC datasets in the PCA space (Figure 7). Harmony is a method that is designed to work specifically with PCA embeddings, and on these 10X datasets it shows a slightly better performance than BATMAN both in terms of iLISI and 10-RNN scores. Both BATMAN and Harmony were able to correctly match the cell populations between the two datasets. In terms of runtime, both tools performed in 14 s.

DISCUSSION

We present BATMAN, a novel method for scRNA-seq dataset integration. It shows significantly better performance than state-of-the-art methods for mixing two datasets in the gene space while efficiently matching cell-type populations and preserving the cell-to-cell relationships of each dataset. It also shows comparable performance against the state-of-the-art tool Harmony in the PCA space. The underlying principle of BATMAN is novel and has never been applied before in scRNA-seq integration. Our parsimonious formulation based on minimum-weight matching not only finds a minimal correction needed to integrate two datasets but also preserves the intrinsic structure of the two datasets.

We show that BATMAN achieves better performance than the state-of-the-art methods in terms of widely used metrics such as LISI on a wide range of datasets in the gene space.

Limitations of the Study

The current implementation of BATMAN works only with two datasets. However, in principle, it can be used to integrate multiple datasets (similar to multiple integration in MNNcorrect).

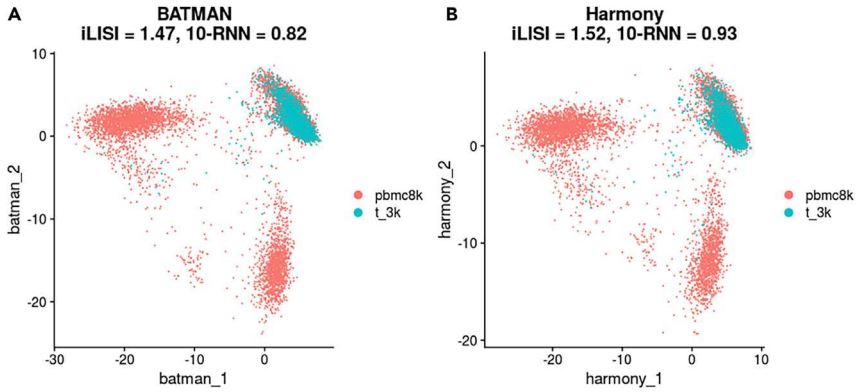


Figure 7. Integration of PBMC 8k and Pan T Cells Datasets (10X Genomics) in the PCA Space
(A) BATMAN; (B) Harmony.

Resources Availability

Lead Contact

Igor Mandric (imandric@ucla.edu)

Materials Availability

This study did not generate new unique reagents.

Data and Code Availability

BATMAN is available at https://github.com/mandricigor/batman.

METHODS

All methods can be found in the accompanying Transparent Methods supplemental file.

SUPPLEMENTAL INFORMATION

Supplemental Information can be found online at https://doi.org/10.1016/j.isci.2020.101185.

ACKNOWLEDGMENTS

E.H., M.T., and B.L.H. were partially supported by the National Science Foundation (Grant No. 1705197). E.H. and M.T. were also partially supported by NIH/NHGRI HG010505-02. E.H., I.M., and M.T. were also partially funded by NIH 1R56MD013312. I.M. and M.K.F. were also partially supported by NIH R01HG009120, NIH R01MH115676, and NIH U01CA194393. E.H. was also partially supported by NIH 1R01MH115979, NIH 5R25GM112625, and NIH 5UL1TR001881.

AUTHOR CONTRIBUTIONS

Conceptualization, E.H., I.M.; Methodology, I.M.; Writing, I.M., B.L.H., M.K.F., M.T., E.H.; Supervision, E.H.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: January 17, 2020
Revised: April 17, 2020
Accepted: May 15, 2020
Published: June 26, 2020

REFERENCES

Amodio, M., van Dijk, D., Srinivasan, K., Chen, W.S., Mohsen, H., Moon, K.R., Campbell, A., Zhao, Y., Wang, X., Venkataswamy, M., et al. (2019). Exploring single-cell data with deep multitasking neural networks. Nat. Methods *16*, 1139–1145.

Angerer, P., Simon, L., Tritschler, S., Wolf, A., Fischer, D., and Theis, F. (2017). Single cells make big data: new challenges and opportunities in transcriptomics. Curr. Opin. Syst. Biol. 85–91.

Brennecke, P., Anders, S., Kim, J.K., Kołodziejczyk, A.A., Zhang, X., Proserpio, V., Baying, B., Benes, V., Teichmann, S.A., Marioni, J.C., and Heisler, M.G. (2013). Accounting for technical noise in single-cell RNA-Seq experiments. Nat. Methods *10*, 1093–1095.

Büttner, M., Miao, Z., Wolf, F.A., Teichmann, S.A., and Theis, F.J. (2019). A test metric for assessing single-cell RNA-Seq batch correction. Nat. Methods *16*, 43–49.

Goh, W.W.B., Wang, W., and Wong, L. (2017). Why batch effects matter in omics data, and how to avoid them. Trends Biotechnol. *35*, 498–507.

Haghverdi, L., Lun, A.T.L., Morgan, M.D., and Marioni, J.C. (2018). Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. Nat. Biotechnol. *36*, 421–427.

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

Johnson, W.E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical Bayes methods. Biostatistics, 118–127.

Korsunsky, I., Millard, N., Fan, J., Slowikowski, K., Zhang, F., Wei, K., Baglaenko, Y., Brenner, M., Loh, P.R., and Raychaudhuri, S. (2019). Fast, sensitive, and accurate integration of single cell data with Harmony. Nat. Methods *36*, 1289–1296.

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

Muraro, M.J., Dharmadhikari, G., Grün, D., Groen, N., Dielen, T., Jansen, E., van Gurp, L., Engelse, M.A., Carlotti, F., de Koning, E.J., and van Oudenaarden, A. (2016). A single-cell transcriptome atlas of the human pancreas. Cell Syst. *3*, 385–394.e3.

Pierson, E., and Yau, C. (2015). ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome Biol. *16*, 241.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.P. (2018). A general and flexible method for signal extraction from single-cell RNA-Seq data. Nat. Commun. *9*, 284.

Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015). Limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Res. *43*, e47.

Segerstolpe, Å., Palasantza, A., Eliasson, P., Andersson, E.M., Andréasson, A.C., Sun, X., Picelli, S., Sabirsh, A., Clausen, M., Bjursell, M.K., et al. (2016). Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. Cell Metab. *24*, 593–607.

Stuart, T., Butler, A., Hoffman, P., Hafemeister, C., Papalexi, E., Mauck, W.M., 3rd, Hao, Y., Stoeckius, M., Smibert, P., and Satija, R. (2019). Comprehensive integration of single-cell data. Cell *177*, 1888–1902.e21.

Svensson, V., Vento-Tormo, R., and Teichmann, S.A. (2018). Exponential scaling of single-cell RNA-seq in the past decade. Nat. Protoc. 599–604.

Thompson, M., Chen, Z.J., Rahmani, E., and Halperin, E. (2019). CONFINED: distinguishing biological from technical sources of variation by leveraging multiple methylation datasets. Genome Biol. *20*, 138.

Wang, T., Johnson, T.S., Shao, W., Lu, Z., Helm, B.R., Zhang, J., and Huang, K. (2019). BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes. Genome Biol. *20*, 165.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. Genome Biol. *18*, 174.

iScience, Volume *23*

Supplemental Information

BATMAN: Fast and Accurate Integration of Single-Cell RNA-Seq Datasets via Minimum-Weight Matching

Igor Mandric, Brian L. Hill, Malika K. Freund, Michael Thompson, and Eran Halperin

Supplemental Information

Transparent Methods

Parsimonious integration of scRNA-Seq datasets

We refer to a scRNA-Seq dataset D as a set of N M-dimensional points where M is the number of genes. Consider a query dataset D_1 and a reference dataset D_2 and assume for simplicity that both D_1 and D_2 consist of the same number of cells. This assumption is necessary only for the sake of formal problem formulation and will be omitted later on. We define the batch effect vector for a cell:

Definition 1. *Suppose that a cell c is sequenced twice (i.e, the same cell sequenced in two different batches, D_1 and D_2) yielding two expression profiles $E^c_{D_1}$ and $E^c_{D_2}$. The batch effect vector for cell c is defined as the vector $B_c = E^c_{D_2} - E^c_{D_1}$.*

We also introduce the notion of scRNA-Seq dataset alignment.

Definition 2. *Given two scRNA-Seq datasets D_1 and D_2 of equal sizes, an alignment of D_1 onto D_2 is a one-to-one correspondence between the cells of D_1 and D_2.*

One would be able to estimate batch effects between two datasets $D_1$and D_2 if they shared a sample of cells. Indeed, according to the Definition 1, one would include RNA of a group of cells twice - once in D_1 and once in D_2- compute batch effect vectors in these cells, and then extrapolate batch effect vectors onto the whole dataset. In reality, sequencing the same cell twice is infeasible (since the cell is destroyed in the sequencing process) and, therefore, it is impossible to directly compute the batch effect vectors of these cells. However, we postulate that if D_1 and D_2 originate from the same biological condition then for each cell $c_1 \in D_1$ there exists a cell $c_2 \in D_2$ such that the expression profile of $c_2 \in D_2$, $E^{c_2}_{D_2}$, is closest to the expression profile of cell $c_1 \in D_1$, $E^{c_1}_{D_1}$, if c_1 were to be sequenced twice (in D_1 and D_2). As we do not know which cell $c_2 \in D_2$ is closest to the unobserved expression profile $E^{c_1}_{D_2}$, we search for such an alignment of D_1 onto D_2 which minimizes the total Euclidean distance between the cells in the dataset D_1 and their corresponding (unobserved) cells in D_2. In order to correct for batch effects, we need to compute the batch effect vector for each cell. Thus, we have to solve the following:

Parsimonious Batch Effect Correction (PBEC) problem. *Given two equal-size scRNA-Seq datasets D_1 and D_2, find an alignment of the dataset D_1 onto D_2 for which the total length of the batch effect vectors across all the cells in D_1 is minimized.*

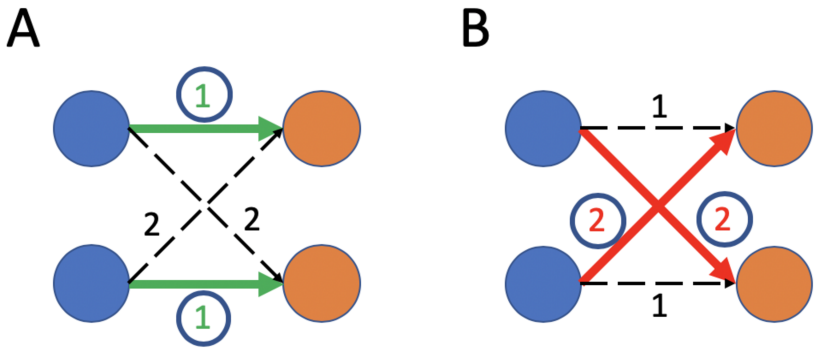


Figure S1: Possible anchor pairs. Related to Figures 1,3,6, and 7. The two datasets (query - blue and reference - orange) consist of two cells. The weights on the edges are the Euclidean distances between the points. There are only two possible anchor pairs. A) The assignment is parsimonious since the total weight of the translation is equal to 2. The local structure of the query dataset is preserved. B) The assignment is not parsimonious (the total weight is 4) and it results in destroying the local structure of the query dataset.

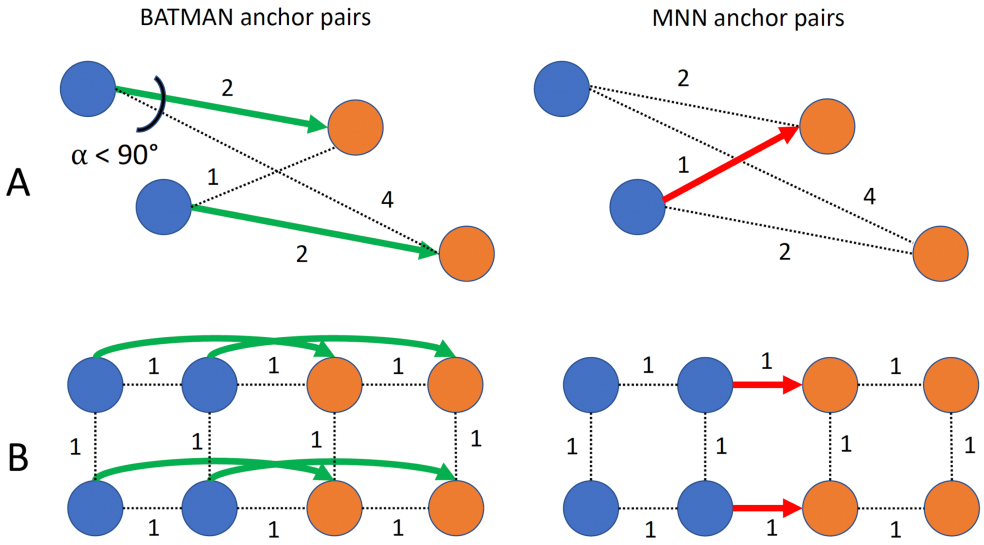


Figure S2: BATMAN anchor pairs vs mutual nearest neighbors anchor pairs. Related to Figures 1,3, 6, and 7. The dotted lines are used to depict the Euclidean distances between the corresponding cells. A) The two datasets consist of two points each (blue and orange), the batch effects are **non-orthogonal** to the biological signal ($\alpha < 90°$). BATMAN identifies the correct anchor pairs, while the MNN approaches fail. B) The two datasets consist of four points each and the batch effects are **large**. BATMAN identifies the correct anchor points, while the MNN approaches fail.

The parsimony of batch effect vectors is strikingly intuitive. To illustrate the idea in a trivial case, suppose that we have two datasets and each dataset consists of two cells (Figure S1). There exist two different alignments between the datasets. If we align the cells as depicted in Figure S1A, the total length of batch effect vectors is equal to 2 (the sum of the green edges). In this case, the local structure of the query dataset is preserved. However, if we align the cells as depicted in Figure S1B, then the total length of batch effect vectors is 4 and the local structure of the query dataset is compromised. This case only helps us illustrate the principle of the parsimonious alignment, and

in order to emphasize its attractiveness over the existing methods we have to consider more challenging scenarios. MNN-based methods (in particular, MNNcorrect (Haghverdi *et al.*, 2018)) fail to properly correct for batch effects in cases when batch effects are not orthogonal to the biological signal or the magnitude of the batch effect vectors is large. Figures S2A and S2B show that parsimonious alignment provides a reasonable solution to the integration problem in these cases, while the mutual nearest neighbor-based approaches fail.

The assumption that the sizes of the datasets D_1 and D_2 are equal is not a realistic one. However, in case of unequal dataset sizes, one can, for example, subsample the larger dataset to the size of the smaller one. In the next section, we present an approach based on selecting an equal number of representative cells in both datasets.

BATMAN: BATch integration via minimum weight MAtchiNg

In this section, we present our approach for solving the PBEC problem called BATMAN (BATch integration via minimum weight MAtchiNg). Suppose that we have two datasets, a query dataset D_1 and a reference dataset D_2. The datasets are assumed to be log-normalized (Luecken and Theis, 2019). First, let us assume that $|D_1| = |D_2|$. Then, solving the PBEC problem can be performed by computing the minimum weight matching in the weighted complete bipartite graph $G = (V = V_{D_1} \cup V_{D_2}, E, w)$ with $w(x, y) = d(x, y), x \in V_1, y \in V_2$, where $d(x, y)$ is the Euclidean distance between gene expression profiles of the cell x in the query dataset and the cell y in the reference dataset. However, in practice, the equinumerosity of D_1 and D_2 is almost never met. Therefore, directly solving PBEC problem on datasets with different number of cells is infeasible. To overcome this issue, we propose a novel algorithm, BATMAN. Instead of matching each cell in the query to a cell in the reference, BATMAN first identifies a set of representative cells in each dataset and then solves PBEC with respect to them. The solution of PBEC on the representative cells from D_1 and D_2 consists of a set of anchor pairs. The anchor pairs are then used to compute batch effect vectors in the representative cells. To determine the batch effect vector in a cell belonging to D_1(not an anchor cell), we compute a weighted average of the batch effect vectors corresponding to the top k closest representative cells in D_1. In more detail, BATMAN consists of the following steps:

1. ***Identification of representative cells.*** Representative cells of an scRNA-Seq dataset are the cells which are located in the high-density regions of the joint gene expression distribution. We propose finding representative cells by using clustering. As clustering in high-dimensional spaces is prone to multiple issues such as the "curse of dimensionality" (Steinbach, Ertöz and Kumar, 2004), we first compute PCA embeddings for each dataset separately. After, we perform clustering (for example, K-means: for small datasets up to 1000 cells, $K \approx 50$; for larger datasets, $K \gtrsim 300$) on each of the two datasets and then identify the cluster centers C_1 and C_2 to use in the original gene space.
2. ***Building the anchor graph.*** We build the weighted complete bipartite graph $G = (C_1 \cup C_2, E, w)$ - the *anchor graph*, where C_1 and C_2 are the cluster centers identified in the previous step of D_1 and D_2 respectively, and the weight of an edge $(x, y) \in E, x \in V_1, y \in V_2$ is equal to the Euclidean distance between the gene expression vectors x and y.
3. ***Computing the minimum weight matching.*** Next, we find the maximum-cardinality minimum weight bipartite matching in the anchor graph G. The endpoints of the edges belonging to the minimum weight matching represent the anchor pairs.

4. ***Computing the batch effect vectors in the representative cells.*** Based on the anchor pairs, we compute the batch effect vectors. Given an anchor pair $(x, y), x \in V_1, y \in V_2$, the corresponding batch effect vector for the cell x is the vector $T_x = y - x$.
5. ***Extrapolation of batch effect vectors and correction.*** For each cell $x \in D_1$, we compute the batch effect vector T_x as a weighted average across the top k closest representative cells $x_1^R, x_2^R, \ldots, x_k^R \in D_1$:

$$T_x = \frac{1}{k} \sum_{i=1}^{k} \alpha_i \, x_i^R \, , \sum_{i=1}^{k} \alpha_i = 1$$

For cell x, we choose α_i to be proportional to $d(x, x_i^R)$. After the batch effect vectorsT_x are determined in each cell x of D_1, we correct for them, i.e. $x \rightarrow x + T_x$.

Dimensionality reduction of the datasets in Step 1 of our algorithm is crucial since it allows us to overcome the "curse of dimensionality" and reduces the runtime of clustering. The number of principal components should be large enough to ensure that the cells are separable by cell types (usually 20-30 (Stuart *et al.*, 2019)). Clustering ensures that the representative cells are distributed throughout the whole volume of the datasets (unlike for MNN-based methods; see Figure S4). Steps 2 and 3 represent the parsimonious alignment of the most representative cells between the two datasets which results in the set of anchor pairs. Finally, in Steps 4 and 5 we compute and correct for batch effects in each cell of the query dataset D_1.

Integrating datasets with multiple cell populations

In the case when the two scRNA-Seq datasets consist of multiple cell types (cell populations), we forbid anchor pairs with different cell type labels. That is, we propose filtering out edges of the anchor graph that are unlikely to be present in the matching solution. To accomplish this, we rely on transcriptional signatures of cell types, capturing the set of genes upregulated in each cell type. Assuming transcriptional signatures are characteristic to each cell type regardless of the technology, batch effects, or any other differences and systematic biases, we expect the correlation between two transcriptional profiles belonging to the same cell type to be high and the correlation of two transcriptional profiles belonging to different cell types to be lower. We use this expectation to filter out unlikely edges in the anchor graph. Namely, if the correlation between gene expression vectors $x \in V_{D_1}$ and $y \in V_{D_2}$ is below a threshold (for example, 0.7), then such an edge is removed from the graph. Filtering not only improves matching of the same cell types between the two datasets, but also significantly reduces the runtime of Step 4 of the BATMAN algorithm as the anchor graph becomes sparser.

Non-concordant clustering between the datasets

Step 1 of BATMAN uses clustering to find the set of representative cells in each batch. A potential pitfall of such an approach is that in the case when the two datasets have different densities of their joint gene expressions as the naive minimum weight matching can fail due to the different sizes of the clusters being matched (Figure S3). In this case, the minimum weight matching can establish correspondences between clusters of significantly different sizes, and batch effects can not be fully

corrected. To overcome this issue, we propose to replicate each representative cell according to the size of the cluster it belongs to. Namely, for larger clusters, we introduce additional representative cells (copies of the existing ones, Figure S3A). We make sure that the number of representative cells in the two datasets is the same. This helps to avoid biases caused by matching of representative cells whose clusters have significantly different sizes (Figure S3B).

Speeding up BATMAN

The steps of BATMAN are not computationally intensive except for Step 4, which involves the computation of the minimum weight matching in the anchor graph for a large enough number of anchors. Despite the fact that polynomial-time algorithms exist for its solving, it still represents a bottleneck for large graphs; for example, the well-known blossom algorithm has complexity $O(N^3)$(Edmonds, 1965; Galil, 1986). An optional speed-up which allows the application of BATMAN to large datasets uses approximation algorithms for the minimum weight matching. The standard greedy algorithm takes only $O(N\ log\ N)$ time (Vazirani, 2003) and, therefore, can be used to identify anchor pairs in very large graphs. However, it can yield a suboptimal solution which is at most twice as bad with respect to the total matching weight of the optimal solution.

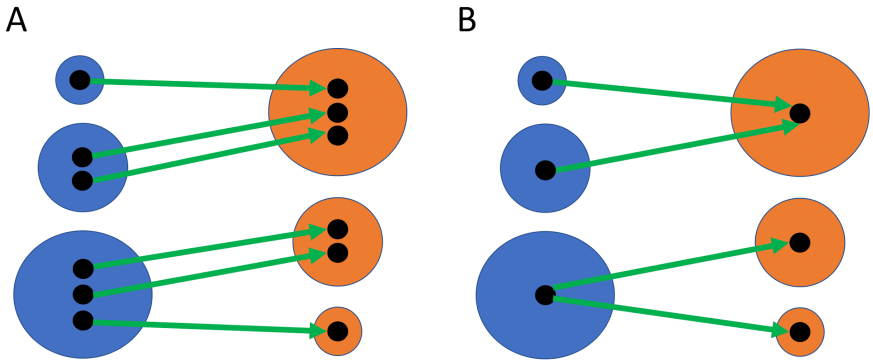


Figure S3: Non-concordant clustering. Related to Figures 1, 3, 6, and 7. A) Introducing additional representative points; B) Final correspondence between clusters.

Supplemental Figures and Tables

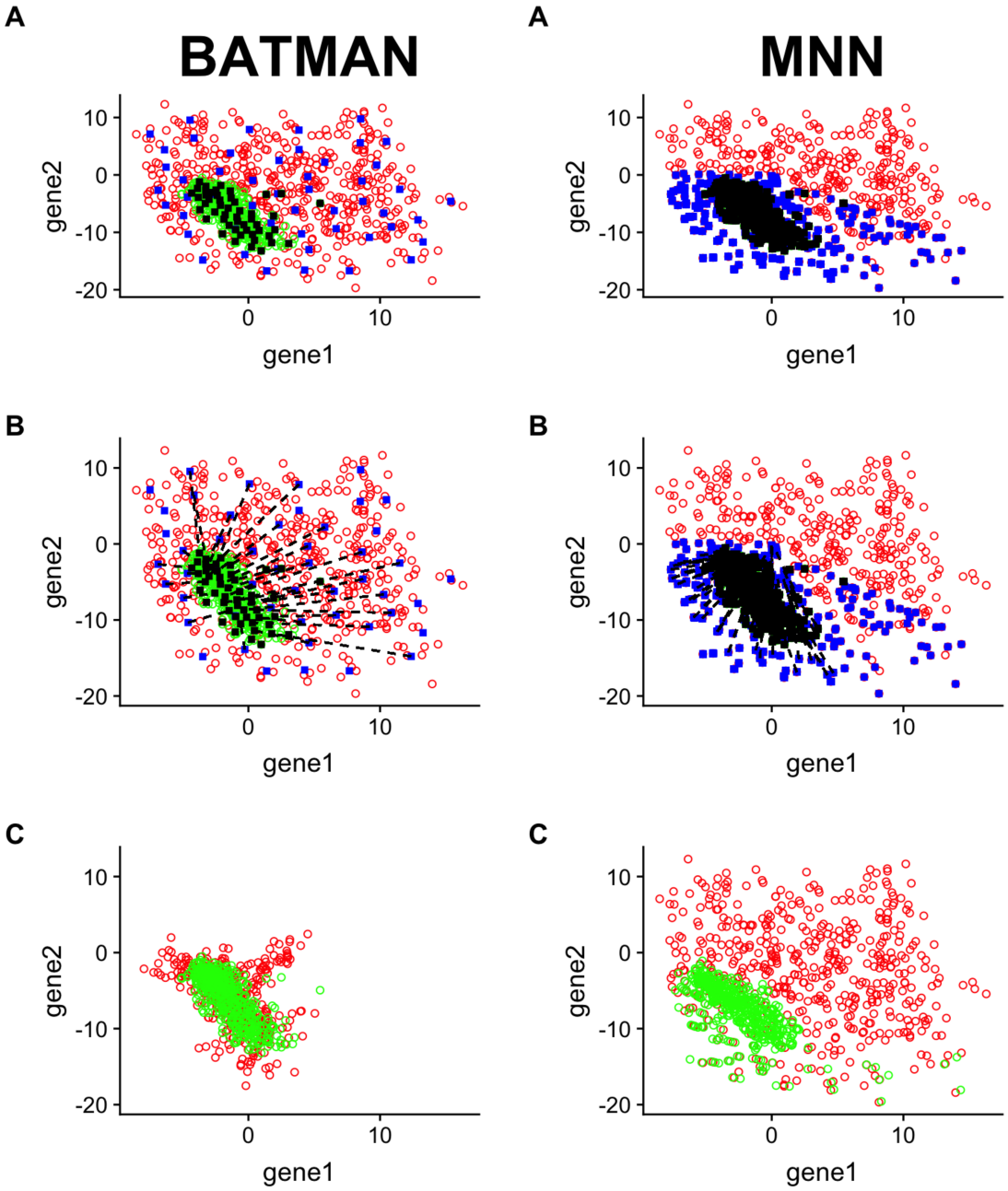


Figure S4: Choosing anchor cells. Related to Figures 1, 3, 6, and 7. (A, B) BATMAN considers anchor cells across all the volume of both datasets, while MNNcorrect (and other MNN-based methods) consider only anchors at the frontier between the datasets. (C) BATMAN successfully integrates two datasets, while MNN-based methods fail.

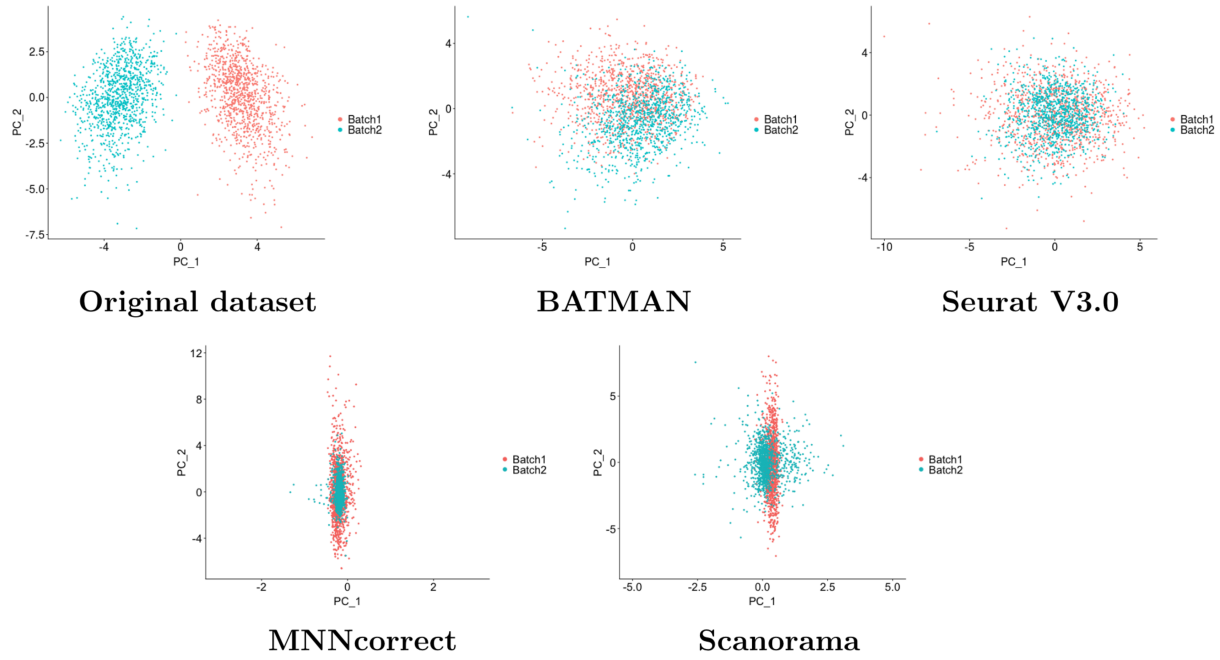


Figure S5: Simulated datasets with large dropout (LB-DR scenario) - PCA plots. Related to Figure 1. Each dataset consists of 1000 cells and 1000 genes. The top 2 PCs are plotted.

Metric	No correction	BATMAN	Seurat V3.0	MNNcorrect	Scanorama
Mean iLISI	1.52	**1.70**	1.00	1.01	1.01
CI iLISI	(1.42, 1.65)	**(1.25, 1.89)**	(1.00, 1.00)	(1.00, 1.01)	(1.00, 1.02)
Mean 50-RNN	1.00	**0.89**	0.59	0.53	0.23
CI 50-RNN	(1.00, 1.00)	**(0.87, 0.90)**	(0.57, 0.60)	(0.53, 0.54)	(0.22, 0.25)

Table S1: Integration results in LB-DR scenario (large dropout): iLISI and 50-RNNscores. Related to Figure 1. The best results are emphasized in bold. CI means confidence interval.

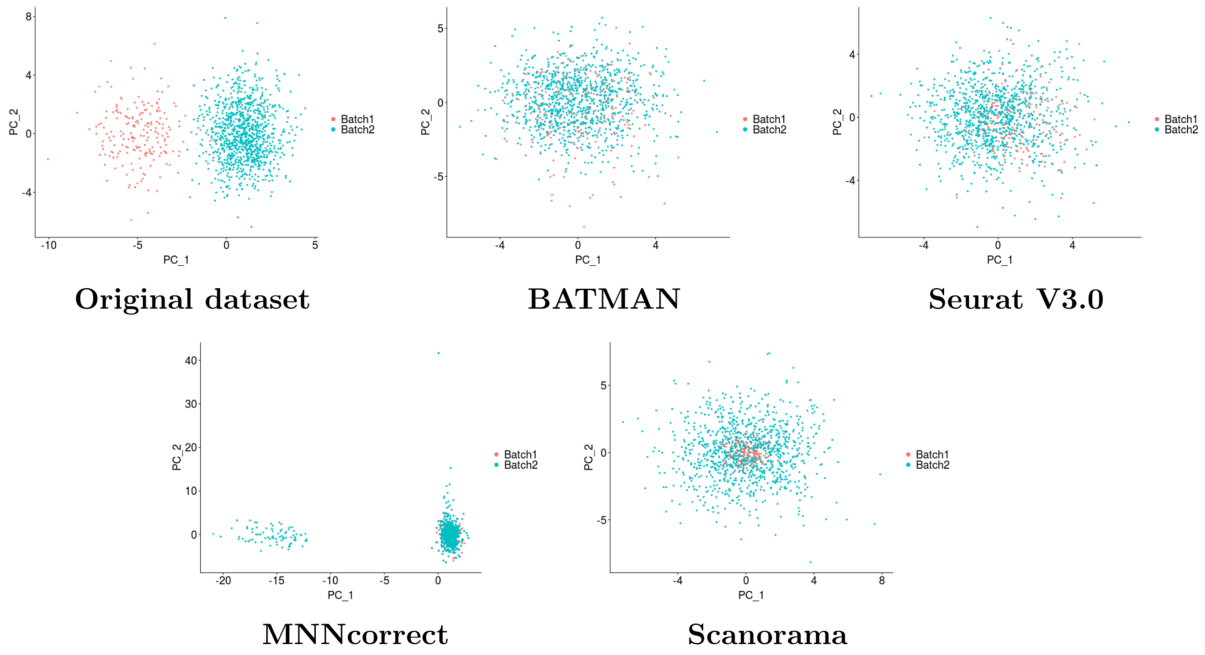


Figure S6: Simulated datasets with unequal batch sizes (LB-UB scenario) - PCA plots. Related to Figure 1. Each dataset consists of 1000 cells and 1000 genes. The top 2 PCs are plotted.

Metric	No correction	BATMAN	Seurat V3.0	MNNcorrect	Scanorama
Mean iLISI	1.31	**1.37**	1.05	1.10	1.00
CI iLISI	(1.06, 1.82)	**(1.16, 1.77)**	(1.00, 1.22)	(1.09, 1.11)	(1.00, 1.00)
Mean 50-RNN	1.00	**0.74**	0.70	0.51	0.25
CI 50-RNN	(1.00, 1.00)	**(0.70, 0.78)**	(0.66, 0.73)	(0.51, 0.51)	(0.22, 0.28)

Table S2: Integration results in LB-UB scenario (large batch effects, unequal batch sizes): iLISI and 50-RNNscores. Related to Figure 1. The best results are emphasized in bold. CI means confidence interval.

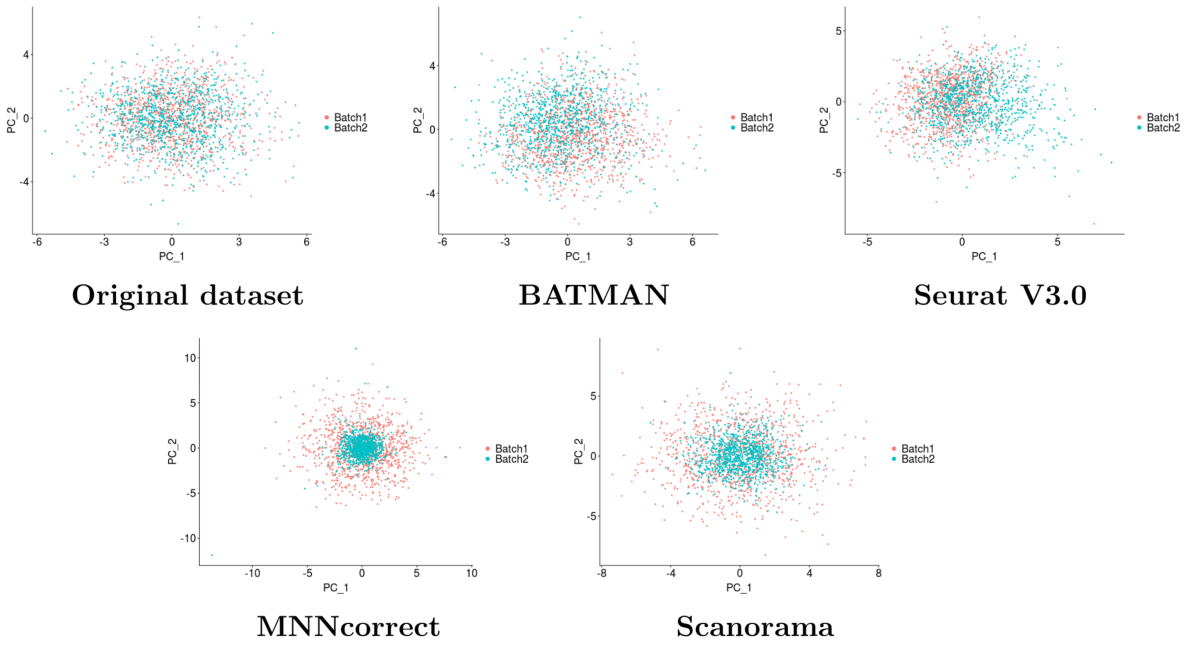


Figure S7: Simulated datasets with small batch sizes (SB scenario) - PCA plots. Related to Figure 1. Each dataset consists of 1000 cells and 1000 genes. The top 2 PCs are plotted.

Metric	Original	BATMAN	Seurat V3.0	MNNcorrect	Scanorama
Mean iLISI	1.88	**1.87**	1.00	1.00	1.00
CI iLISI	(1.70, 1.94)	**(1.66, 1.93)**	(1.00, 1.01)	(1.00, 1.00)	(1.00, 1.01)
Mean 50-RNN	1.00	**0.93**	0.58	0.51	0.13
CI 50-RNN	(1.00, 1.00)	**(0.91, 0.95)**	(0.56, 0.60)	(0.51, 0.52)	(0.11, 0.16)

Table S3: Integration results in SB scenario (small batch effects, unequal batch sizes): iLISI and 50-RNNscores. Related to Figure 1. The best results are emphasized in bold. CI means confidence interval.

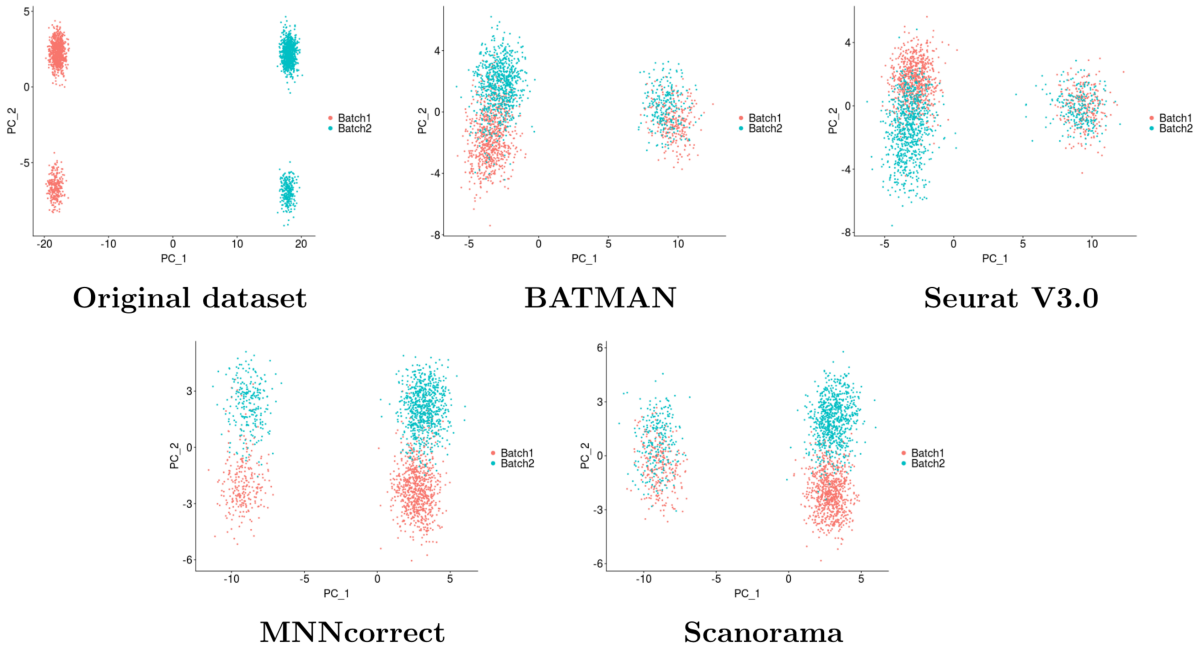


Figure S8: Simulated datasets with large batch effects and two cell types (LB-CT scenario) - PCA plots. Related to Figure 1. Each dataset consists of 1000 cells and 1000 genes. Cell type frequencies are 80% and 20%. The top 2 PCs are plotted.

Metric	Original	BATMAN	Seurat V3.0	MNNcorrect	Scanorama
Mean iLISI	1.00	1.79	1.00	**1.85**	1.61
CI iLISI	(1.00, 1.00)	(1.60, 1.87)	(1.00, 1.00)	**(1.85, 1.85)**	(1.61, 1.61)
Mean cLISI	1.01	1.04	1.03	**1.00**	**1.00**
CI cLISI	(1.01, 1.01)	(1.00, 1.22)	(1.03, 1.03)	**(1.00, 1.00)**	(1.00, 1.00)
Mean 50-RNN	1.00	0.95	0.64	**0.99**	0.96
CI 50-RNN	(1.00, 1.00)	(0.89, 0.98)	(0.64, 0.64)	(0.99, 0.99)	(0.96, 0.96)

Table S4: Integration results in LB-CT scenario (large batch effects, two cell types): iLISI and 50-RNN scores. Related to Figure 1. The best results are emphasized in bold. CI means confidence interval.

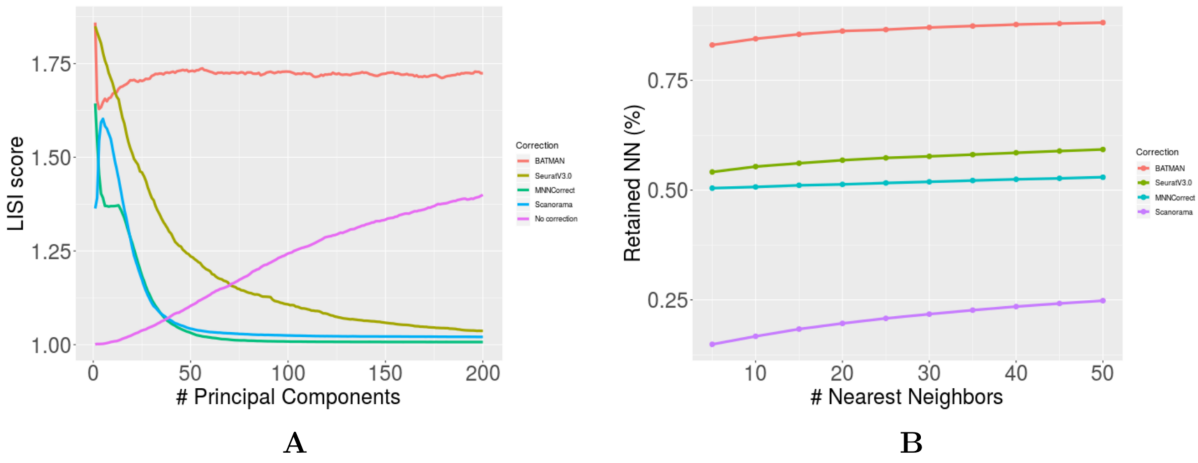


Figure S9: Evaluation of integration with large batch effects and large dropout (LB-DR scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) *k*-RNN score for different values of *k*.

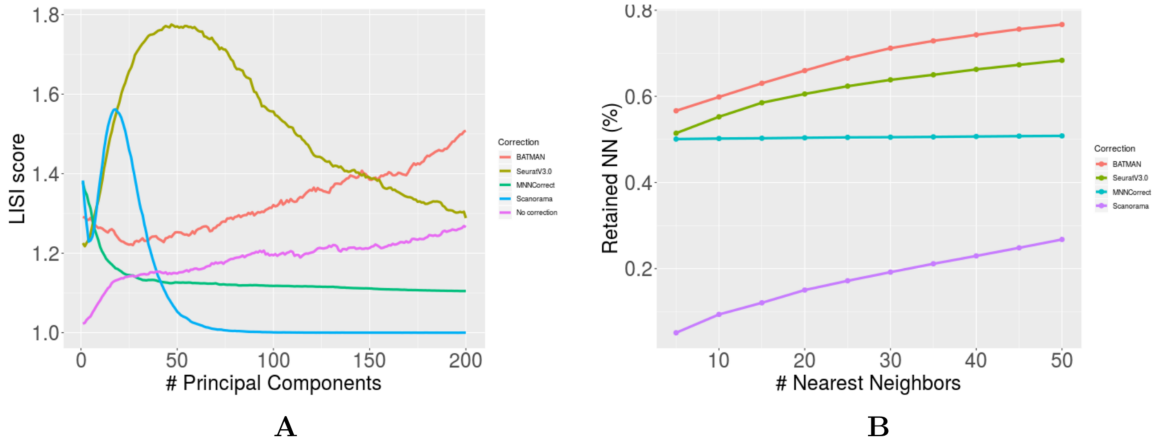


Figure S10: Evaluation of integration with large batch effects and unequal batch sizes (LB-UB scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) *k*-RNN score for different values of *k*.

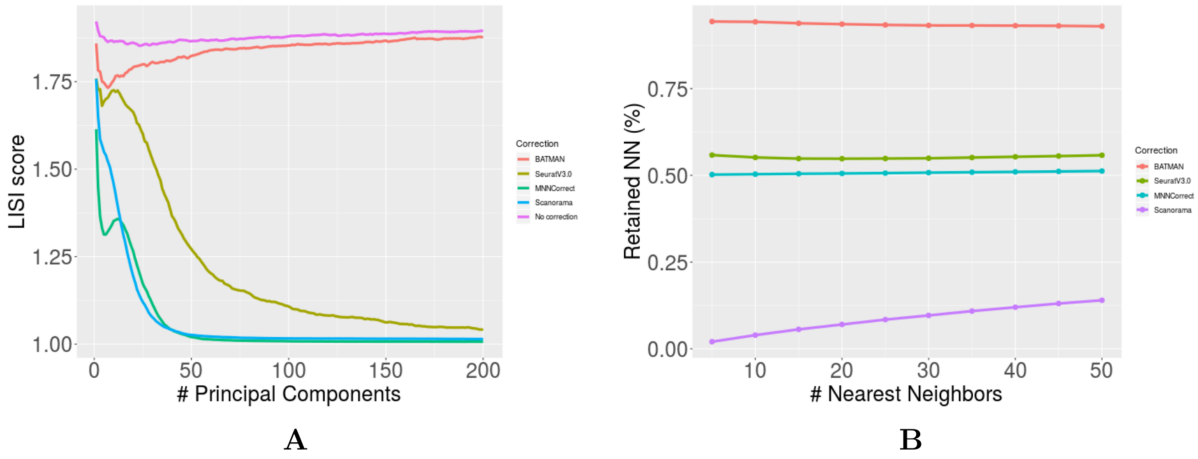


Figure S11: Evaluation of integration with small batch effects (SB scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) *k*-RNN score for different values of *k*.

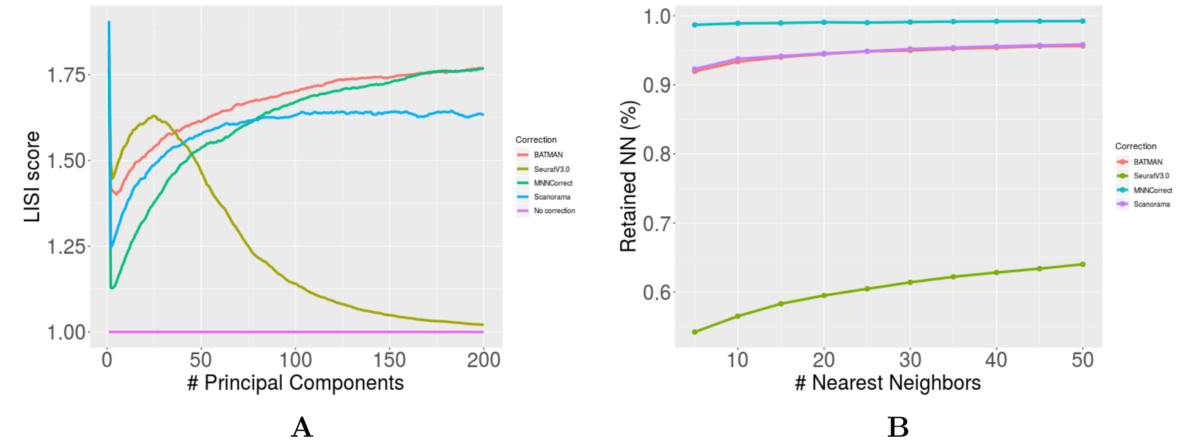


Figure S12: Evaluation of integration with large batch effects and two cell types (LB-CT scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) *k*-RNN score for different values of *k*.

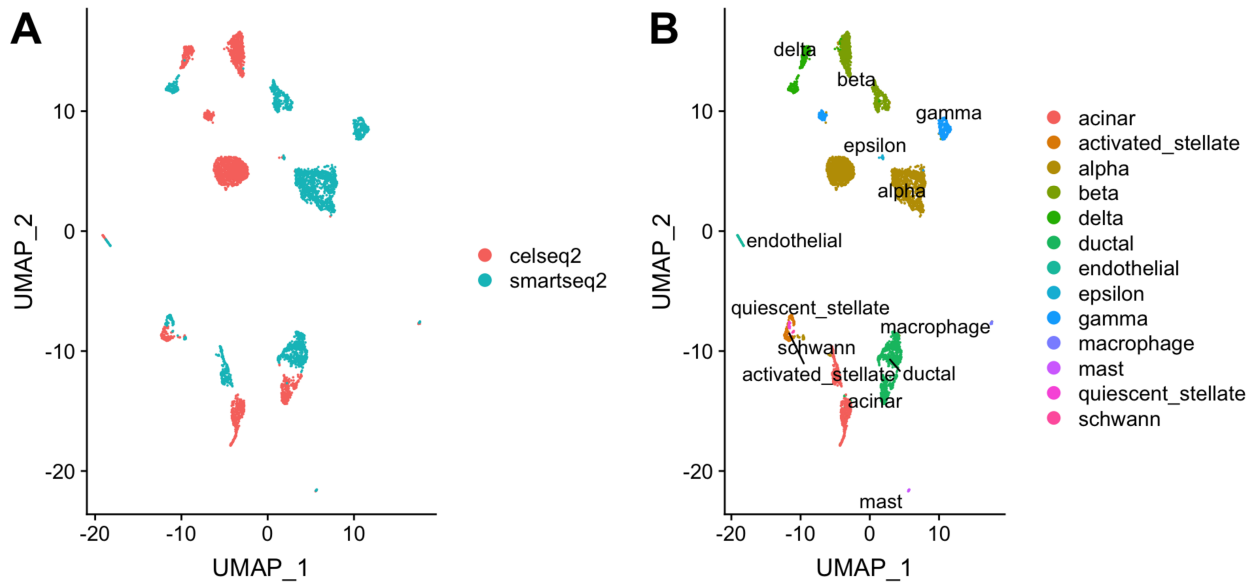


Figure S13: UMAP plots of two pancreatic datasets - CEL-Seq2 and Smart-Seq2. Related to Figure 3. A) Datasets are colored by batch label; B) Datasets are colored by cell types.

Dataset	acinar	stellate	alpha	beta	delta	ductal	endo	epsilon	gamma	macro	mast	q-stellate	schwann
CEL-Seq2	12	4	37	19.5	9	11	1	0.2	4.8	0.7	0.3	0.5	0.2
Smart-Seq2	8	2.3	42	13	5.3	18.5	0.9	0.3	9	0.3	0.3	0.3	0.1

Table S5: Cell-type composition of the two pancreas datasets (in %). Related to Figure 3.

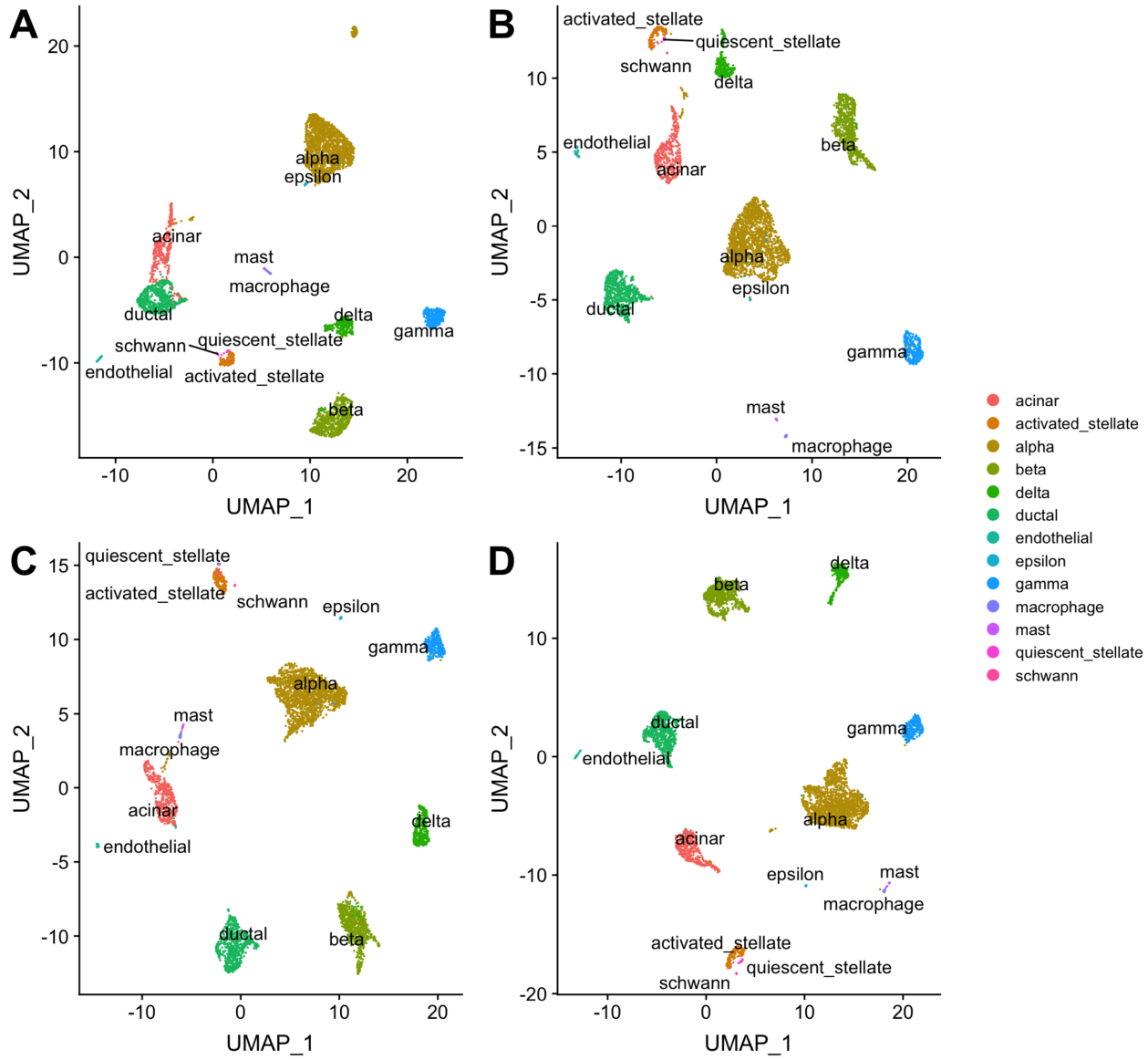


Figure S14: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results in "all cell types" experiment. Related to Figure 3. Datasets are colored by cell types. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.

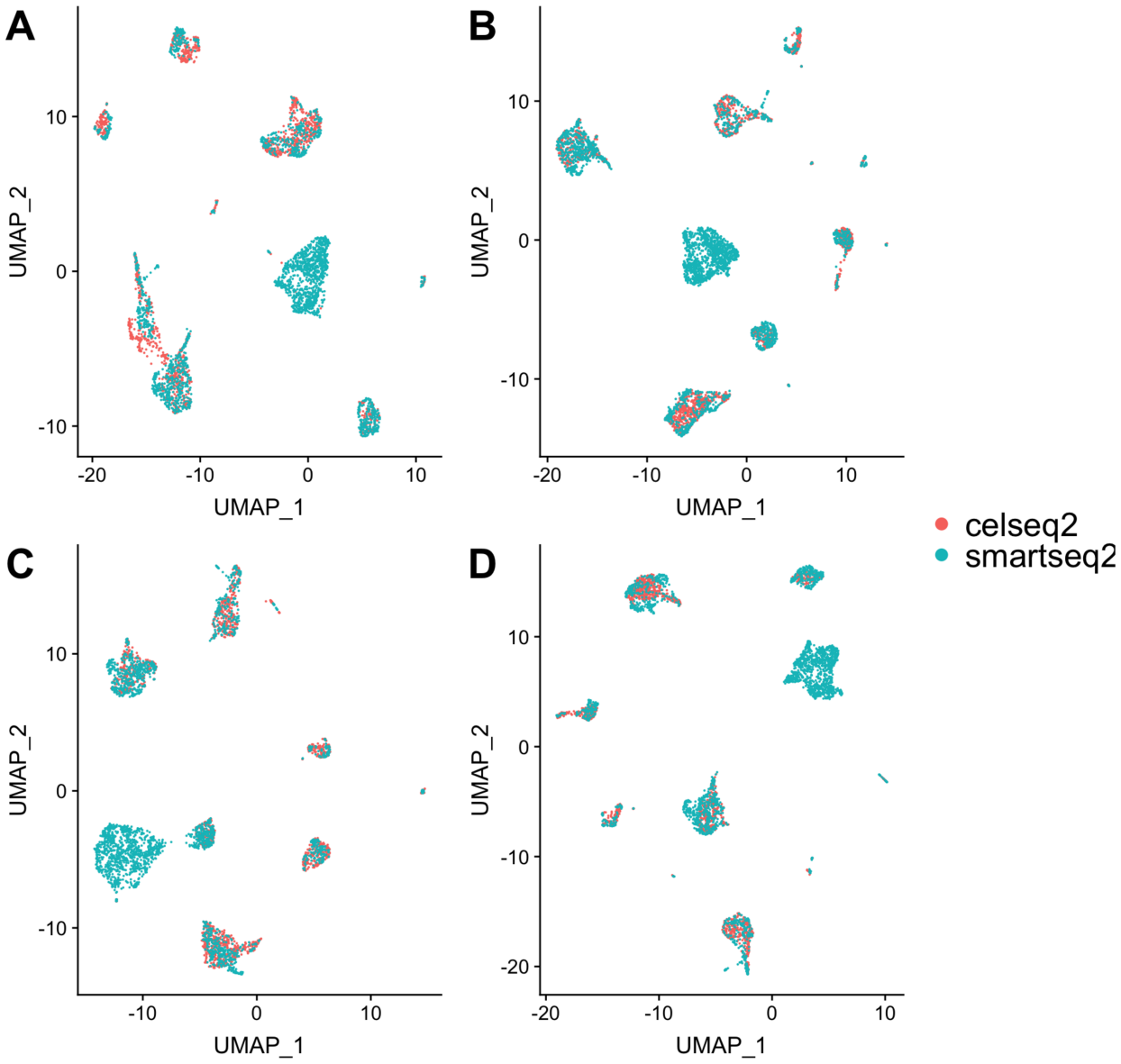


Figure S15: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results in "1-held out" experiment. Related to Figure 3. Datasets are colored by batches. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.

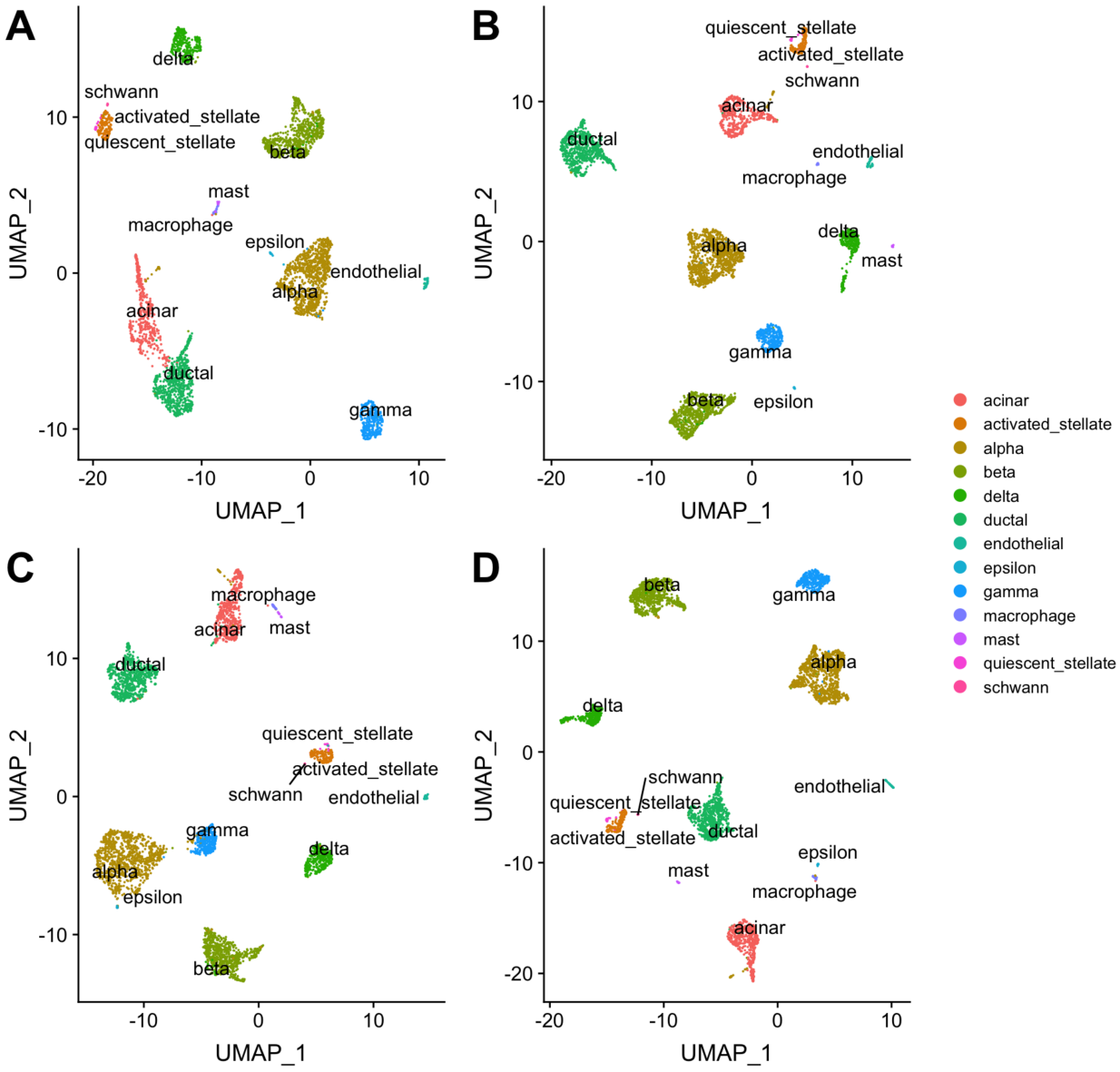


Figure S16: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results in "1-held-out" experiment. Related to Figure 3. Datasets are colored by cell types. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.

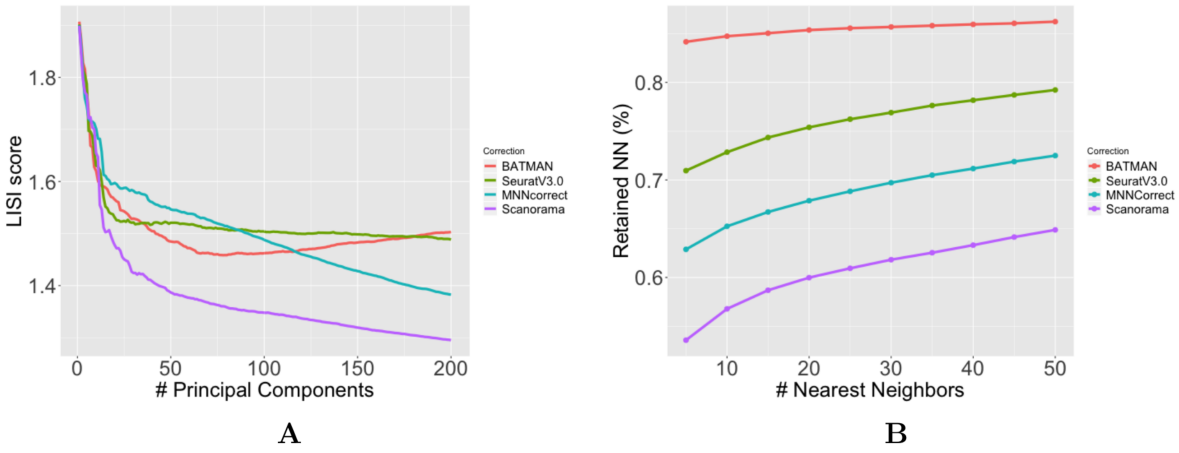


Figure S17: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results. Related to Figure 4. Datasets are colored by cell types. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.