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 $E_{D_1}^{c}$ and $E_{D_2}^{c}$. The batch effect vector for cell c is defined as the vector $B_c = E_{D_2}^{c} - E_{D_1}^{c}$.*

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 $c_2 \in D_2$, $E_{D_2}^{c_2}$, is closest to the expression profile of cell $c_1 \in D_1$, $E_{D_1}^{c_1}$, 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 $E_{D_2}^{c_1}$, 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.*

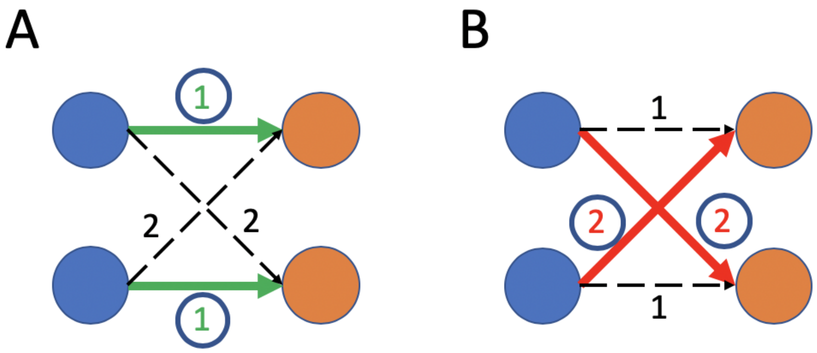


Figure S1: Possible anchor pairs. Related to Figures 1,3,6, and 7. The two datasets (query - blue and reference - orange) consist of two cells. The weights on the edges are the Euclidean distances between the points. There are only two possible anchor pairs. A) The assignment is parsimonious since the 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.

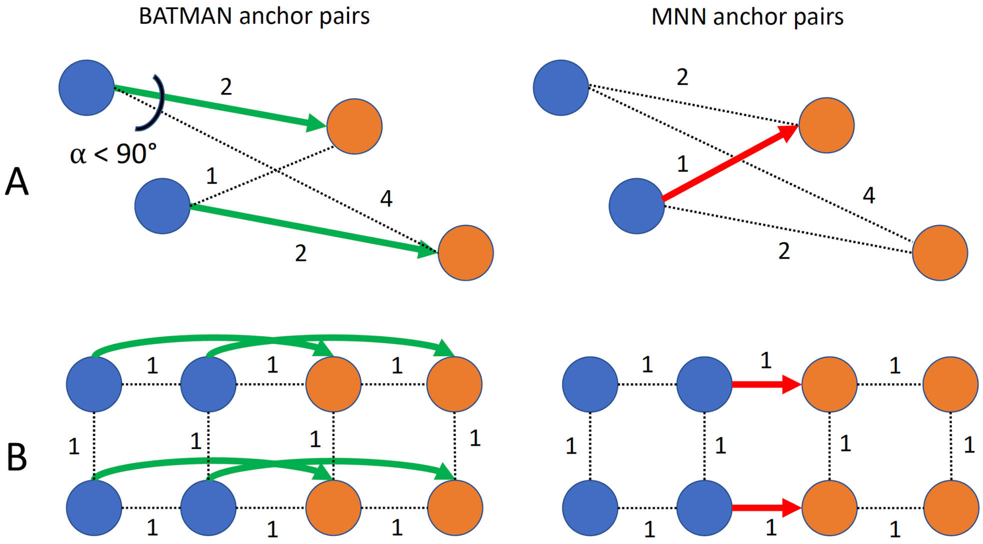


Figure S2: BATMAN anchor pairs vs mutual nearest neighbors anchor pairs. Related to Figures 1,3, 6, and 7. The dotted lines are used to depict the Euclidean distances between the corresponding cells. A) The two datasets consist of two points each (blue and orange), the batch effects are **non-orthogonal** 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.

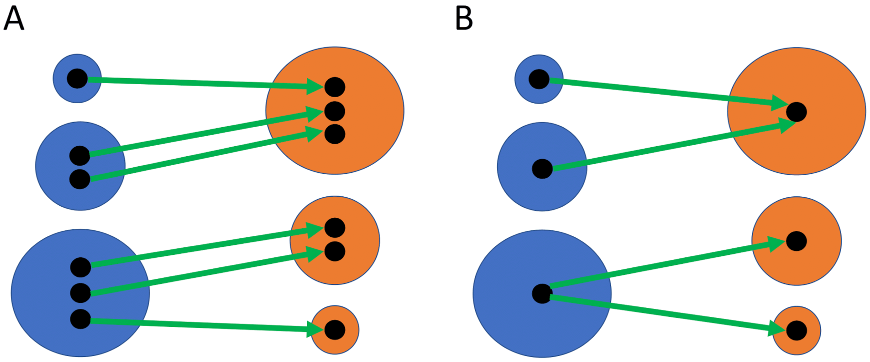


Figure S3: Non-concordant clustering. Related to Figures 1, 3, 6, and 7. A) Introducing additional representative points; B) Final correspondence between clusters.

Supplemental Figures and Tables

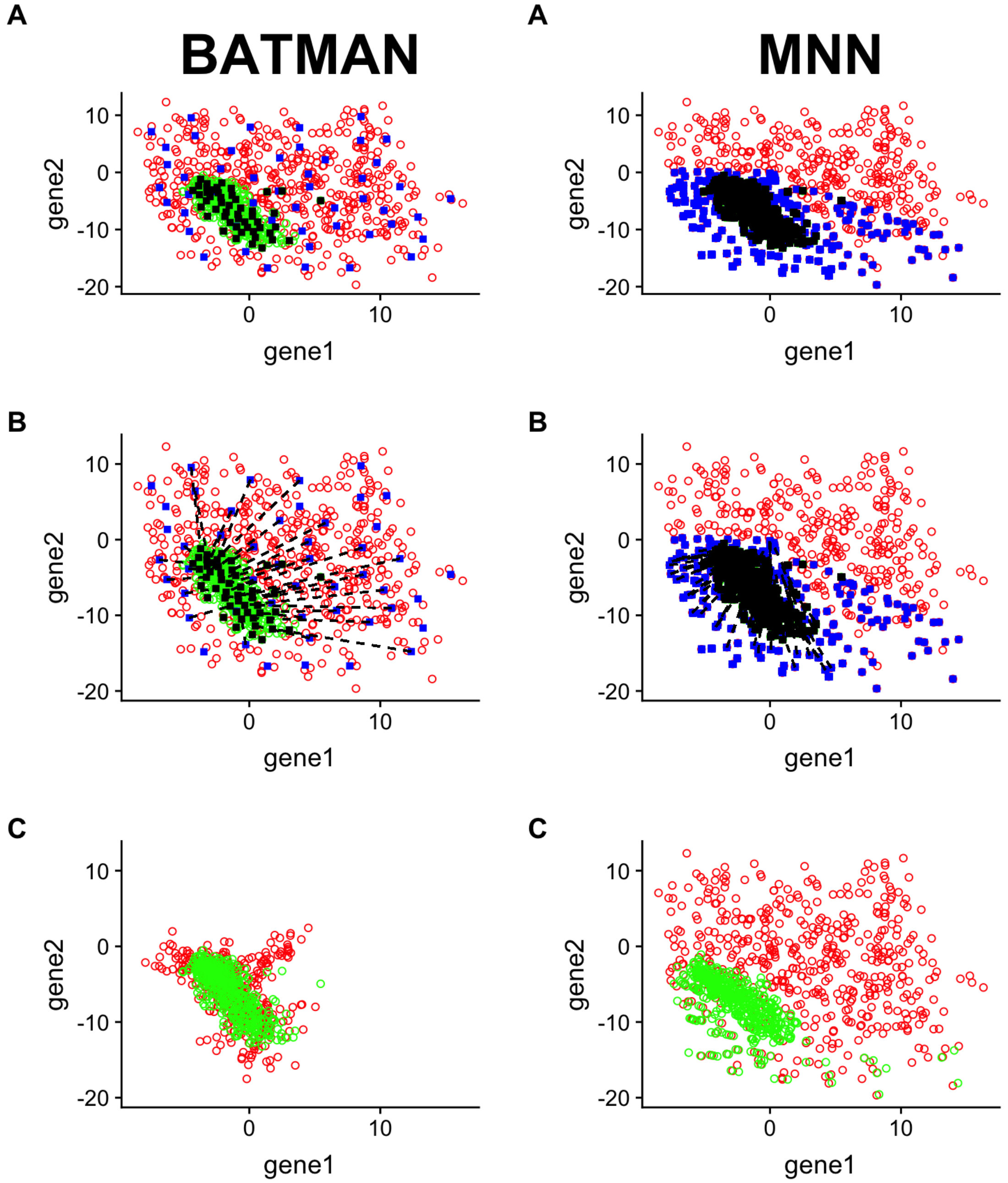
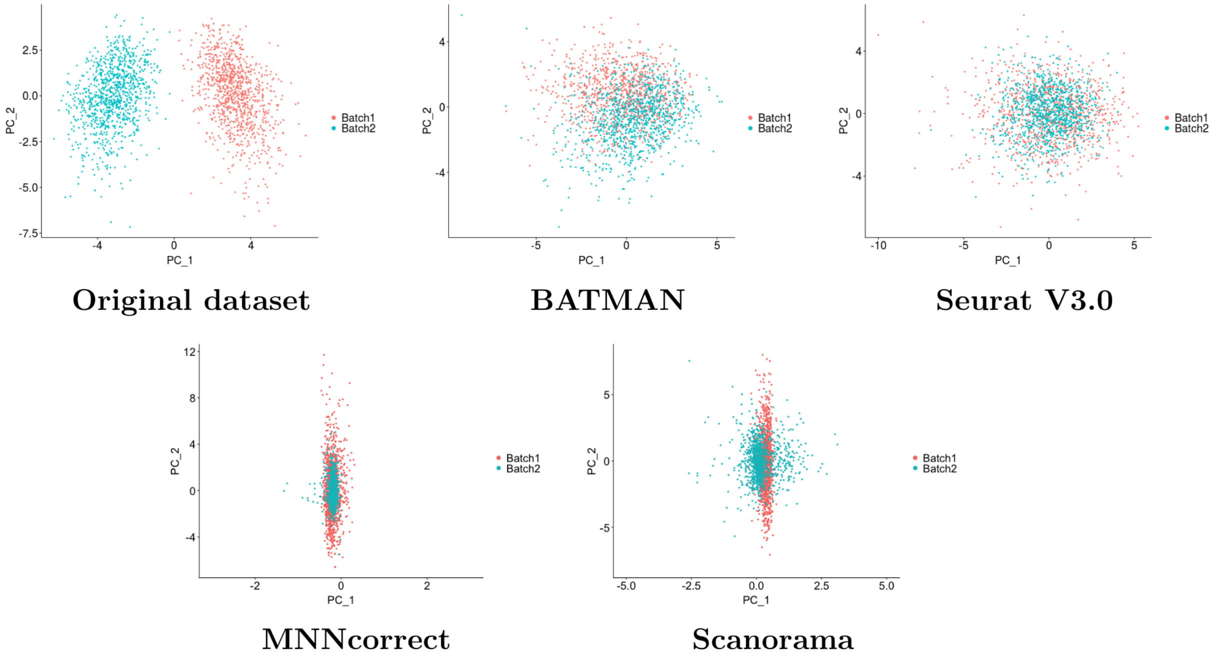


Figure S4: Choosing anchor cells. Related to Figures 1, 3, 6, and 7. (A, B) BATMAN considers anchor cells across all the volume of both datasets, while MNNcorrect (and other MNN-based methods) consider only anchors at the frontier between the datasets. (C) BATMAN successfully integrates two datasets,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



Original dataset

BATMAN

Seurat V3.0

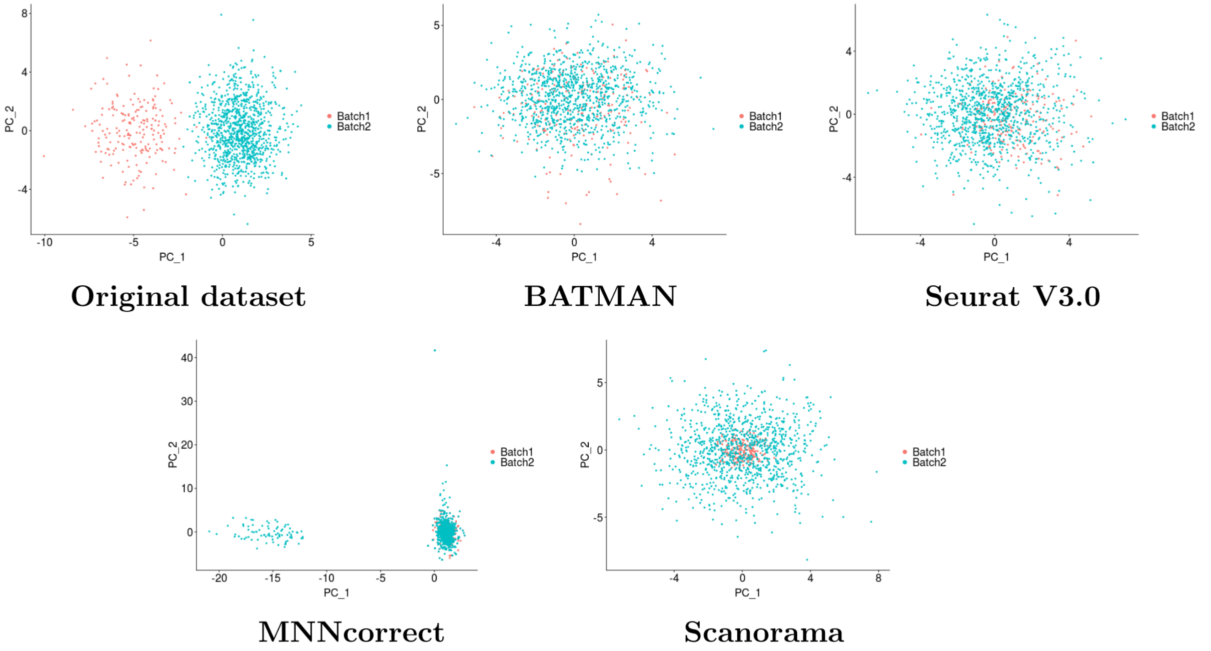
MNNcorrect

Scanorama

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.



Original dataset

BATMAN

Seurat V3.0

MNNcorrect

Scanorama

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.

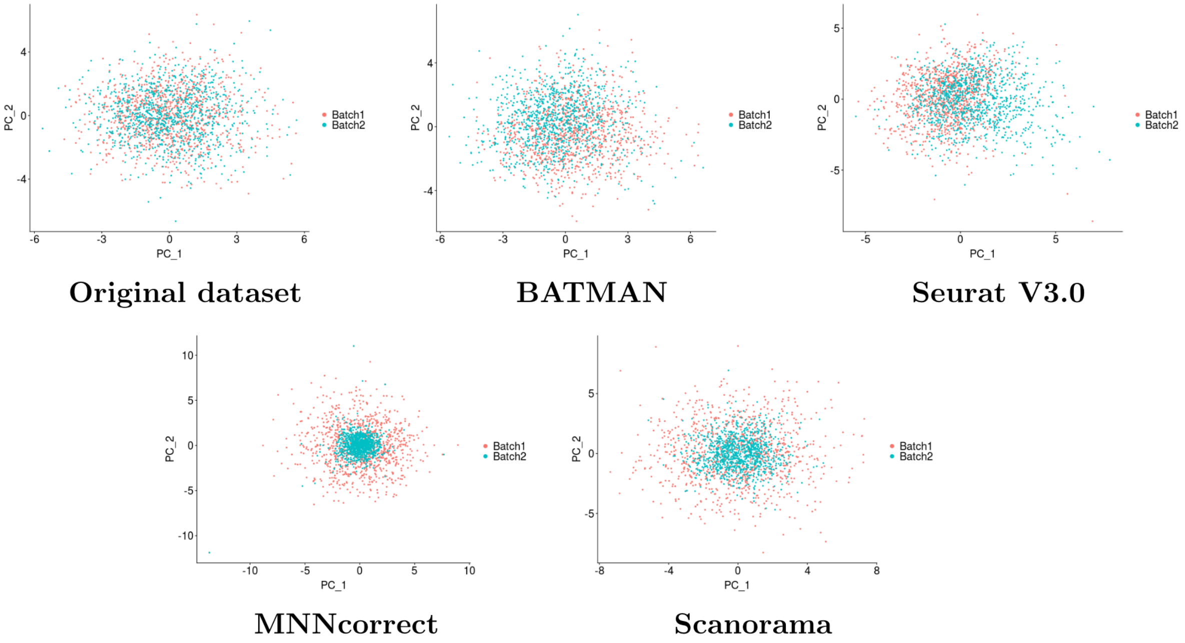


Figure S7: Simulated datasets with small batch sizes (SB scenario) - PCA plots. Related to Figure 1. Each dataset consists of 1000 cells and 1000 genes. The top 2 PCs are plotted.

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.

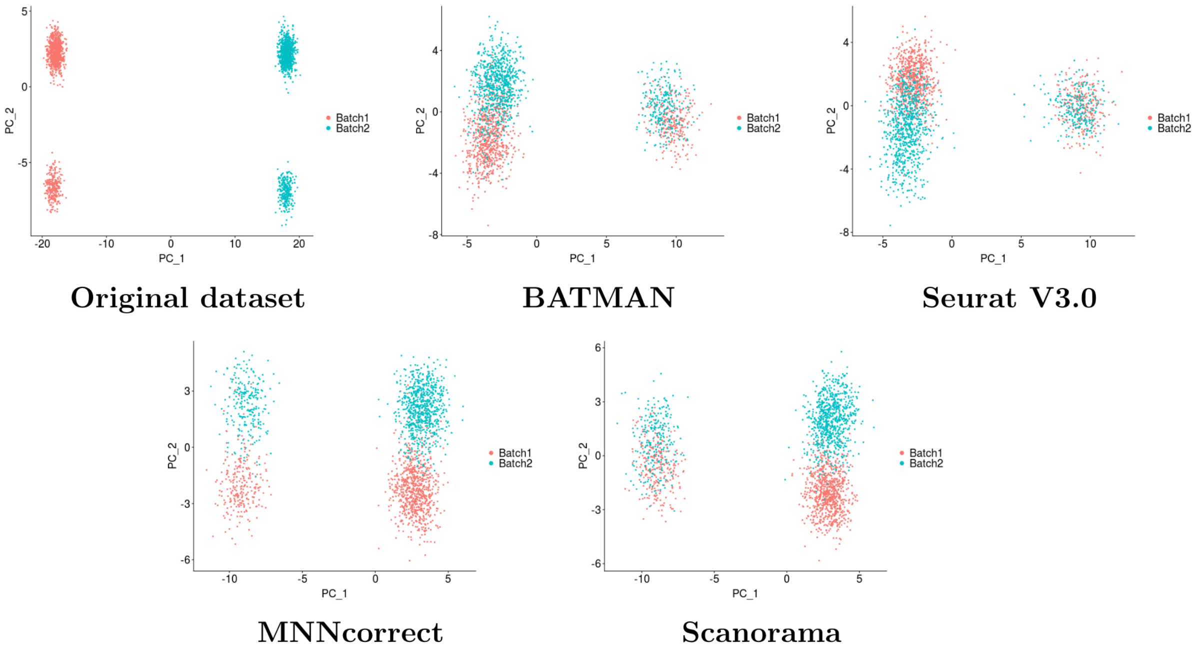


Figure S8: Simulated datasets with large batch effects and two cell types (LB-CT scenario) - PCA plots. Related to Figure 1. Each dataset consists of 1000 cells and 1000 genes. Cell type frequencies are 80% and 20%. The top 2 PCs are plotted.

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.

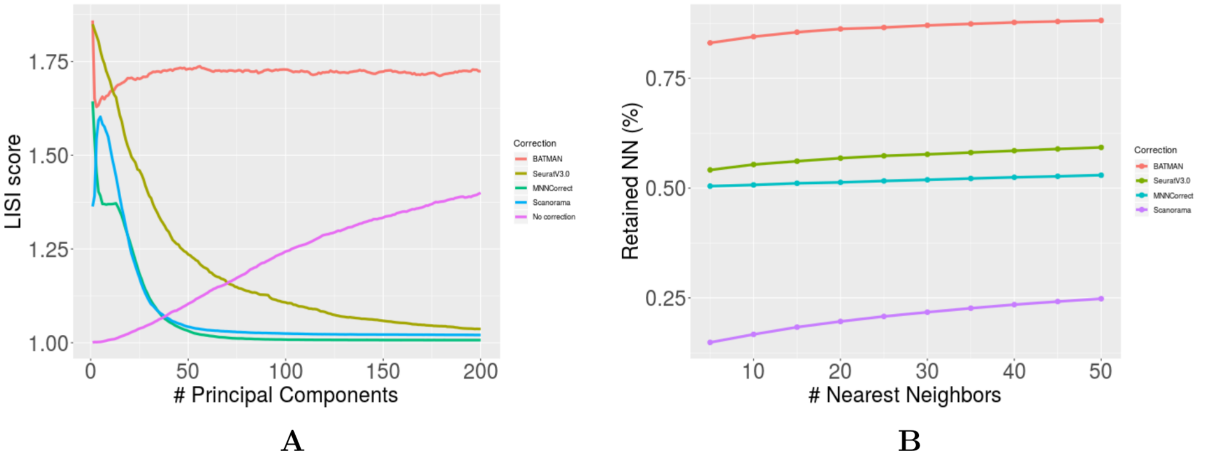


Figure S9: Evaluation of integration with large batch effects and large dropout (LB-DR scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) k-RNN score for different values of k.

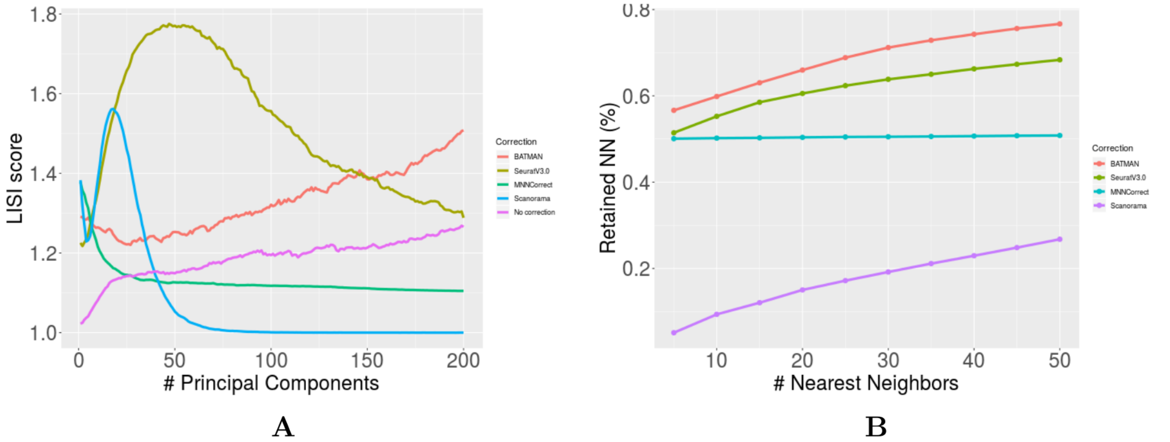


Figure S10: Evaluation of integration with large batch effects and unequal batch sizes (LB-UB scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) k-RNN score for different values of k.

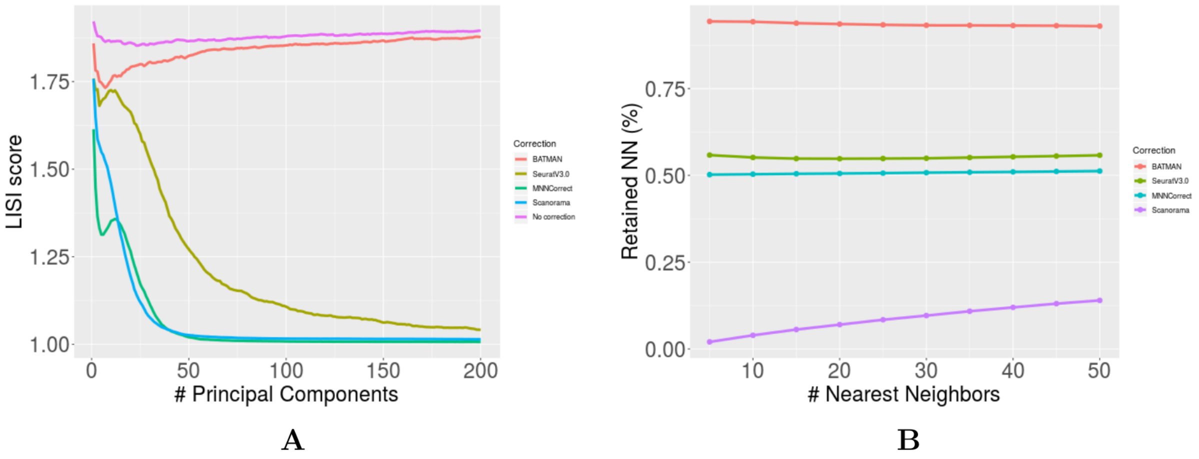


Figure S11: Evaluation of integration with small batch effects (SB scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) k-RNN score for different values of k.

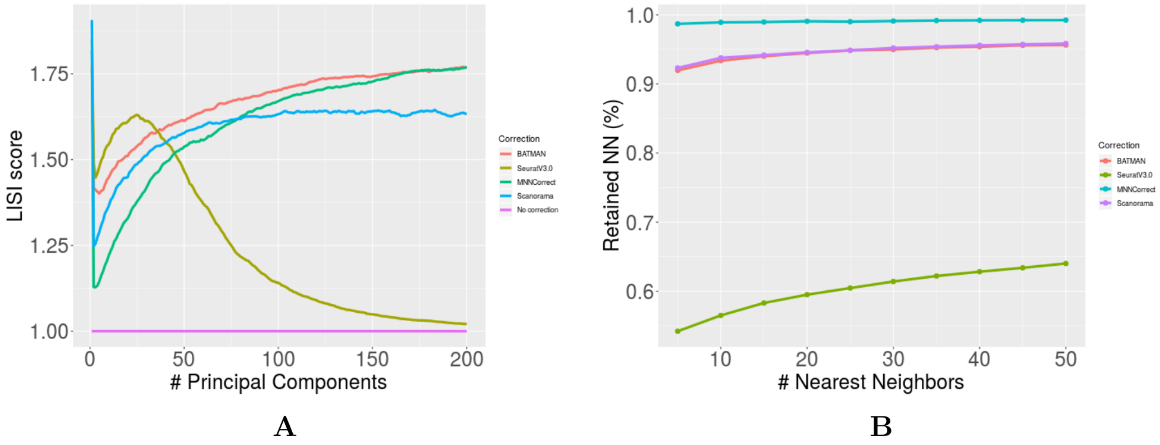


Figure S12: Evaluation of integration with large batch effects and two cell types (LB-CT scenario). Related to Figure 2. A) iLISI score as a function of the number of top principal components. B) k-RNN score for different values of k.

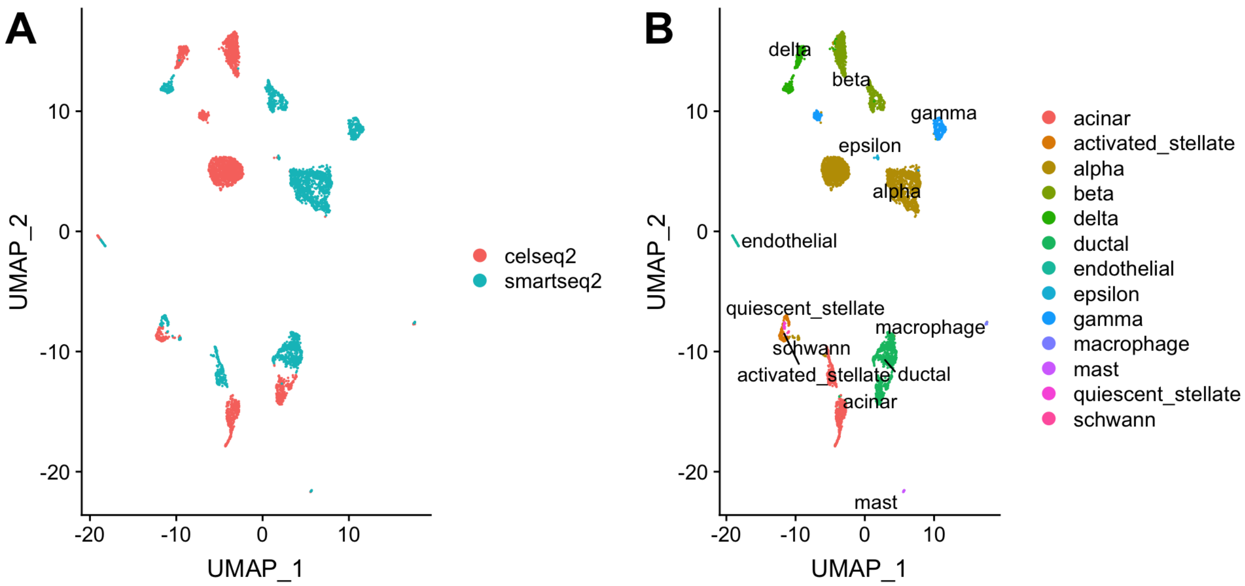


Figure S13: UMAP plots of two pancreatic datasets - CEL-Seq2 and Smart-Seq2. Related to Figure 3. A) Datasets are colored by batch label; B) Datasets are colored by cell types.

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.

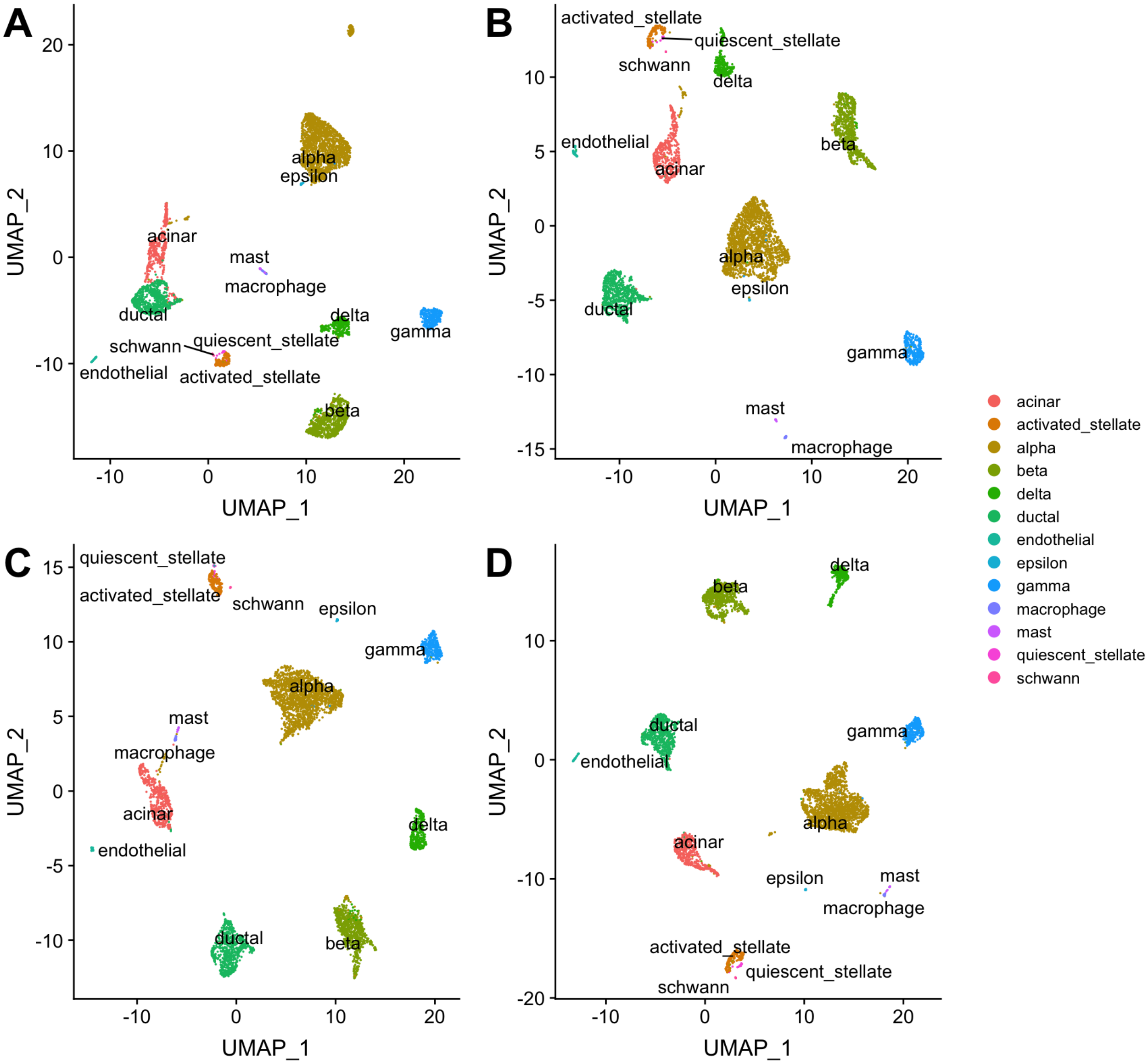


Figure S14: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results in “all cell types” experiment. Related to Figure 3. Datasets are colored by cell types. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.

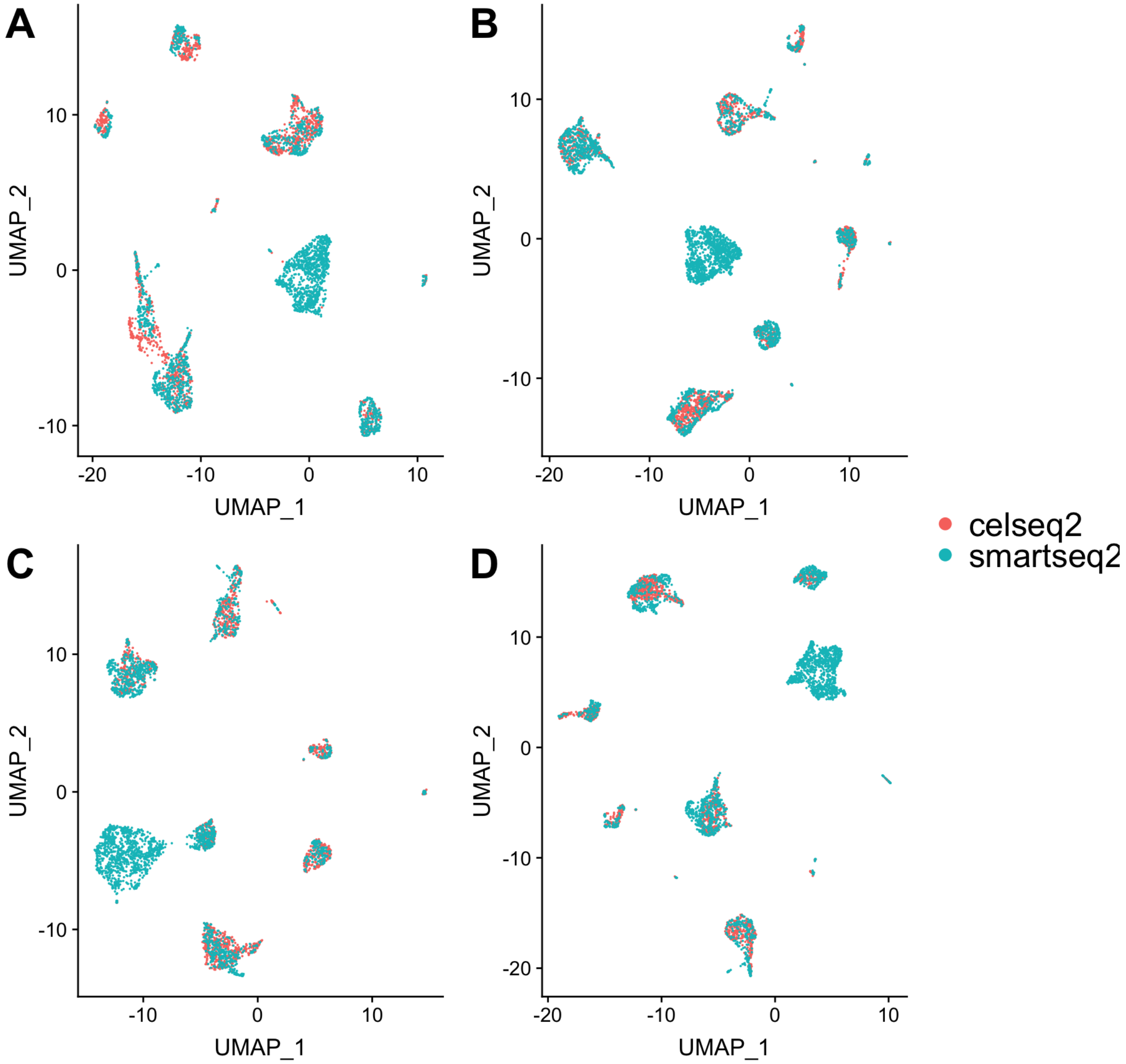


Figure S15: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results in "1-held out" experiment. Related to Figure 3. Datasets are colored by batches. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.

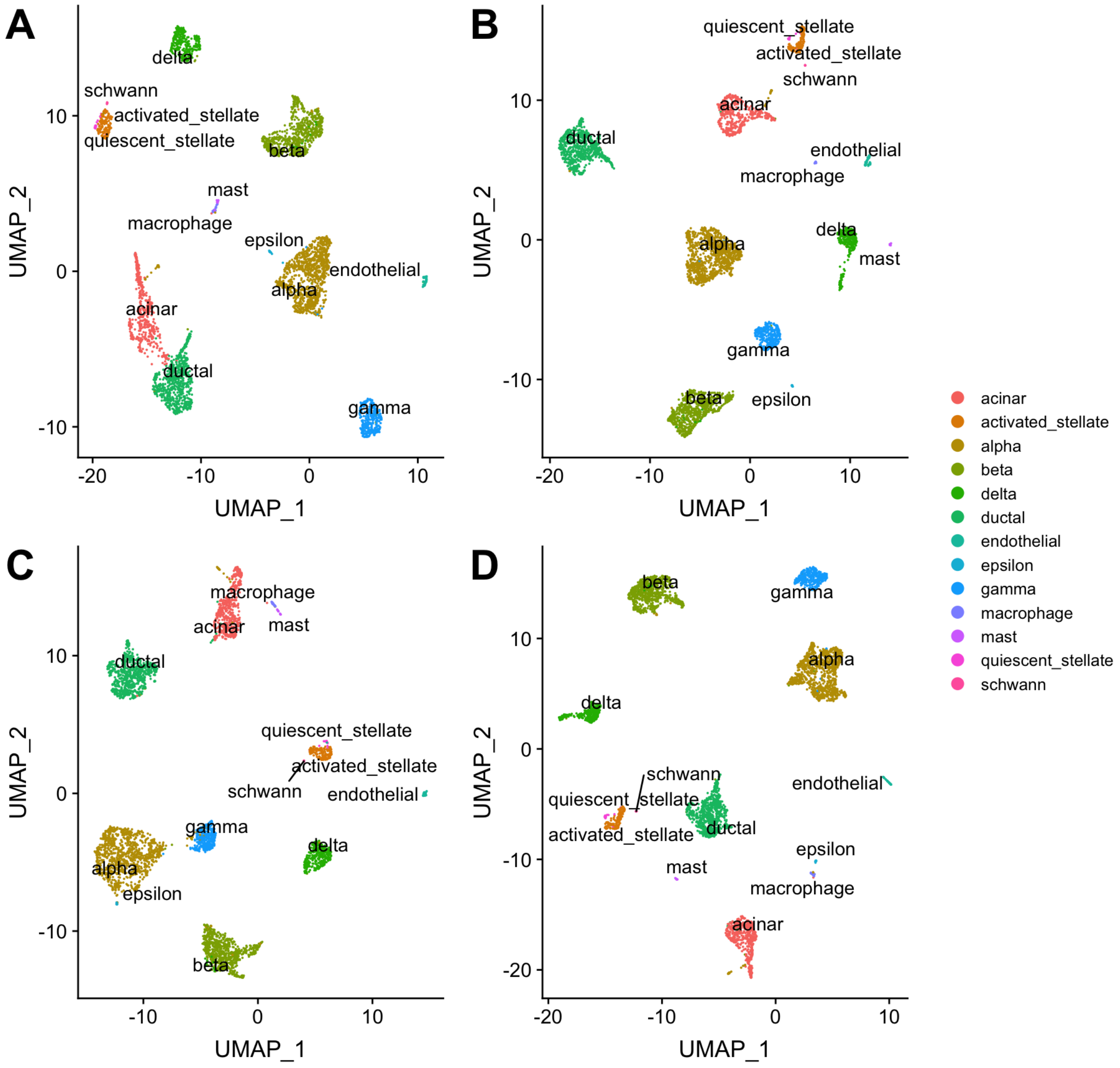


Figure S16: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results in "1-held-out" experiment. Related to Figure 3. Datasets are colored by cell types. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.

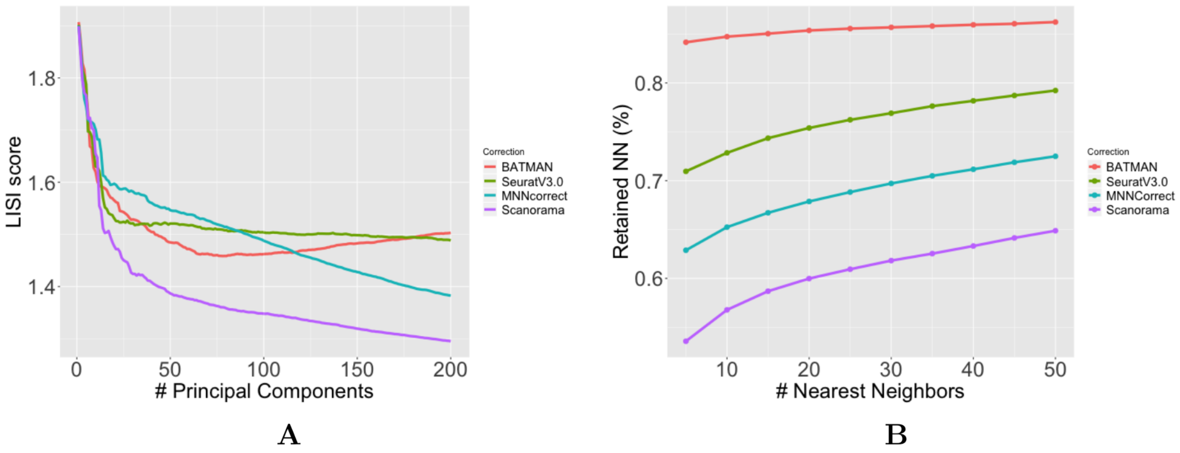


Figure S17: UMAP plots of two pancreatic datasets - CelSeq2 and SmartSeq2: integration results. Related to Figure 4. Datasets are colored by cell types. A) BATMAN; B) Seurat V3.0; C) MNNcorrect; D) Scanorama.