Briefings in Bioinformatics, 00(00), 2020, 1–11

doi: 10.1093/bib/bbaa097
Problem Solving Protocol

SMNN: batch effect correction for single-cell RNA-seq data via supervised mutual nearest neighbor detection

Yuchen Yang†, Gang Li†, Huijun Qian, Kirk C. Wilhelmsen, Yin Shen and Yun Li

Corresponding author: Yun Li. Department of Genetics, Biostatistics and Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA. Fax: (919) 843-4682; E-mail: yunli@med.unc.edu
†These authors contributed equally to this work.

Abstract

Batch effect correction has been recognized to be indispensable when integrating single-cell RNA sequencing (scRNA-seq) data from multiple batches. State-of-the-art methods ignore single-cell cluster label information, but such information can improve the effectiveness of batch effect correction, particularly under realistic scenarios where biological differences are not orthogonal to batch effects. To address this issue, we propose SMNN for batch effect correction of scRNA-seq data via supervised mutual nearest neighbor detection. Our extensive evaluations in simulated and real datasets show that SMNN provides improved merging within the corresponding cell types across batches, leading to reduced differentiation across batches over MNN, Seurat v3 and LIGER. Furthermore, SMNN retains more cell-type-specific features, partially manifested by differentially expressed genes identified between cell types after SMNN correction being biologically more relevant, with precision improving by up to 841.0%.

Key words: single-cell RNA sequencing; batch effect; supervised mutual nearest neighbor

Introduction

An ever-increasing amount of single cell RNA-sequencing (scRNA-seq) data has been generated as scRNA-seq technologies mature and sequencing costs continue dropping. However, large-scale scRNA-seq data, for example those profiling tens of thousands to millions of cells (such as the Human Cell Atlas Project) [1], almost inevitably involve multiple batches across time points, laboratories or experimental protocols. The presence of batch effect renders joint analysis across batches challenging [2, 3]. Batch effect or systematic differences in gene expression profiles across batches not only can obscure the true underlying biology but also may lead to spurious findings. Thus, batch effect correction, which aims to mitigate the discrepancies across batches, is crucial and deemed indispensable for the analysis of scRNA-seq data across batches [4].

Because of its importance, a number of batch effects correction methods has been recently proposed and implemented. Most of these methods, including limma [5], ComBat [6] and

Yuchen Yang is a postdoctoral research fellow in the Department of Genetics at the University of North Carolina at Chapel Hill.
Gang Li is a PhD candidate in the Department of Statistics and Operations Research at the University of North Carolina at Chapel Hill.
Huijun Qian was a PhD student in the Department of Statistics and Operations Research at the University of North Carolina at Chapel Hill.
Kirk C. Wilhelmsen is a Professor in the Departments of Genetics and Neurology at the University of North Carolina at Chapel Hill.
Yin Shen is an Assistant Professor in the Institute for Human Genetics and Department of Neurology at the University of California San Francisco.
Yun Li is an Associated Professor in the Departments of Genetics, Biostatistics and Computer Science at the University of North Carolina at Chapel Hill.
Submitted: 25 January 2020; **Received (in revised form):** 20 April 2020

svaseq [7], are regression-based. Among them, limma and ComBat explicitly model known batch effect as a blocking term. Because of the regression framework adopted, standard statistical approaches to estimate the regression coefficients corresponding to the blocking term can be conveniently employed. In contrast, svaseq is often used to detect the underlying unknown factors of variation, for instance, unrecorded differences in the experimental protocols. Svaseq first identifies these unknown factors as surrogate variables and subsequently corrects them. For these regression-based methods, once the regression coefficients are estimated or the unknown factors are identified, one can then regress out these batch effects accordingly, obtaining residuals that will serve as the batch-effect corrected expression matrix for further analyses. These methods have become standard practice in the analysis of bulk RNA-seq data. However, when it comes to scRNA-seq data, one key underlying assumption behind these methods, in which the cell composition within each batch is identical, might not hold. Consequently, estimates of the coefficients might be inaccurate. As a matter of fact, when applied to scRNA-seq data, the corrected results derived from these methods widely adopted for bulk RNA-seq data might be even inferior to raw data without no correction, in some extreme cases [8].

To address the heterogeneity and high dimensionality of complex data, several dimension-reduction approaches have been adopted. An incomplete list of these strategies includes principal component analysis (PCA), autoencoder or force-based methods such as t-distributed stochastic neighbor embedding (t-SNE) [9]. Through those dimension reduction techniques, one can project new data onto the reference dataset using a set of landmarks from [8, 10–12] to remove batch effects between any new dataset and the reference dataset. Such projection methods require the reference batch that contains all the cell types across batches. As one example, Spitzer *et al.* [11] employed force-based dimension reduction and showed that leveraging a few landmark cell types from bone marrow (the most appropriate tissue in that it provides the most complete coverage of immune cell types) allowed mapping and comparing immune cells across different tissues and species. When applied to scRNA-seq data, however, these methods suffer when cells from a new batch fall out of the space inferred from the reference. Furthermore, determining the dimensionality of the low dimensional manifolds is still an open and challenging problem. To address the limitations of existing methods, two recently developed batch effect correction methods, MNN and Seurat v3, adopt the concept of leveraging information of mutual nearest neighbors (MNNs) across batches [8, 12] and demonstrate superior performance over alternative methods [8, 12]. However, this MNN-based strategy ignores cell-type information and suffers from potentially mismatching cells from different cell types/states across batches, which may lead to undesired correction results. For example, under the scenario depicted in Figure 1b, MNN leads to cluster 1 (C1) and cluster 2 (C2) mis-corrected due to mismatching single cells in the two clusters/cell-types across batches.

To address the above issue, here, we present SMNN, a supervised machine learning method that explicitly incorporates cell-type information. SMNN performs nearest neighbor searching within the same cell type, instead of global searching ignoring cell-type labels (Figure 1a). Cell-type information, when unknown *a priori*, can be inferred via clustering methods [13–16].

Results

SMNN framework

The motivation behind our SMNN is that single-cell cluster or cell-type information has the potential aid the identification of most relevant nearest neighbors and subsequently improves batch effect correction. A preliminary clustering before any correction can provide knowledge regarding cell composition within each batch, which serves as the cellular correspondence across batches (Figure 1a). With this clustering information, we can refine the nearest neighbor searching space within a certain population of cells that are of the same or similar cell type(s) or state(s) across all batches.

SMNN takes a natural two-step approach to leverage cell-type label information for enhanced batch effect correction (Figure 1c and Supplementary Section 1). First, it takes the expression matrices across multiple batches as input and performs clustering separately for each batch. Specifically, in this first step, SMNN uses Seurat v3 [17] where dimension reduction is conducted via PCA to the default of 20 PCs, and then graph-based clustering follows on the dimension-reduced data with resolution parameter of 0.9 [18, 19]. Obtaining an accurate matching of the cluster labels across batches is of paramount importance for subsequent nearest neighbor detection. SMNN requires users to specify a list of marker genes and their corresponding cell-type labels to match clusters/cell types across batches. We, hereafter, refer to this cell type or cluster matching as cluster harmonization across batches. Because not all cell types are necessarily shared across batches, and no prior knowledge exists regarding the exact composition of cell types in each batch, SMNN allows users to take discretion in terms of the marker genes to include, representing the cell types that are believed to be shared across batches. Based on the marker gene information, a harmonized label is assigned to every cluster identified across all the batches according to two criteria: the percentage of cells in a cluster expressing a certain marker gene and the average gene expression levels across all the cells in the cluster. After harmonization, cluster labels are unified across batches. This completes step one of SMNN. Note that if users have a priori knowledge regarding the cluster/cell-type labels, the clustering step could be bypassed completely.

With the harmonized cluster or cell-type label information obtained in the first step, SMNN, in the second step, searches MNNs only within each matched cell type between the first batch (which serves as the reference batch) and any of the other batches (the current batch) and performs batch effect correction accordingly. Compared with MNN or Seurat v3, where the MNNs or anchor cells are searched globally, SMNN identifies neighbors from the same cell population or state. After MNNs are identified, similar to MNN, SMNN first computes batch effect correction vector for each identified pair of cells and then calculates, for each cell, the cell-specific correction vectors by exploiting a Gaussian kernel to obtain a weighted average across all the pair-specific vectors with MNNs of the cell under consideration. The correction vectors obtained from shared cell-types will be applied to correct all cells including those belonging to batch-specific cell types (detailed in Supplementary Section 2). Each cell's correction vector is further scaled according to the cell's location in the space defined by the correction vector and standardized according to quantiles across batches, in order to eliminate 'kissing effects'. 'Kissing effects' refer to the phenomenon that only the surfaces of cell-clouds

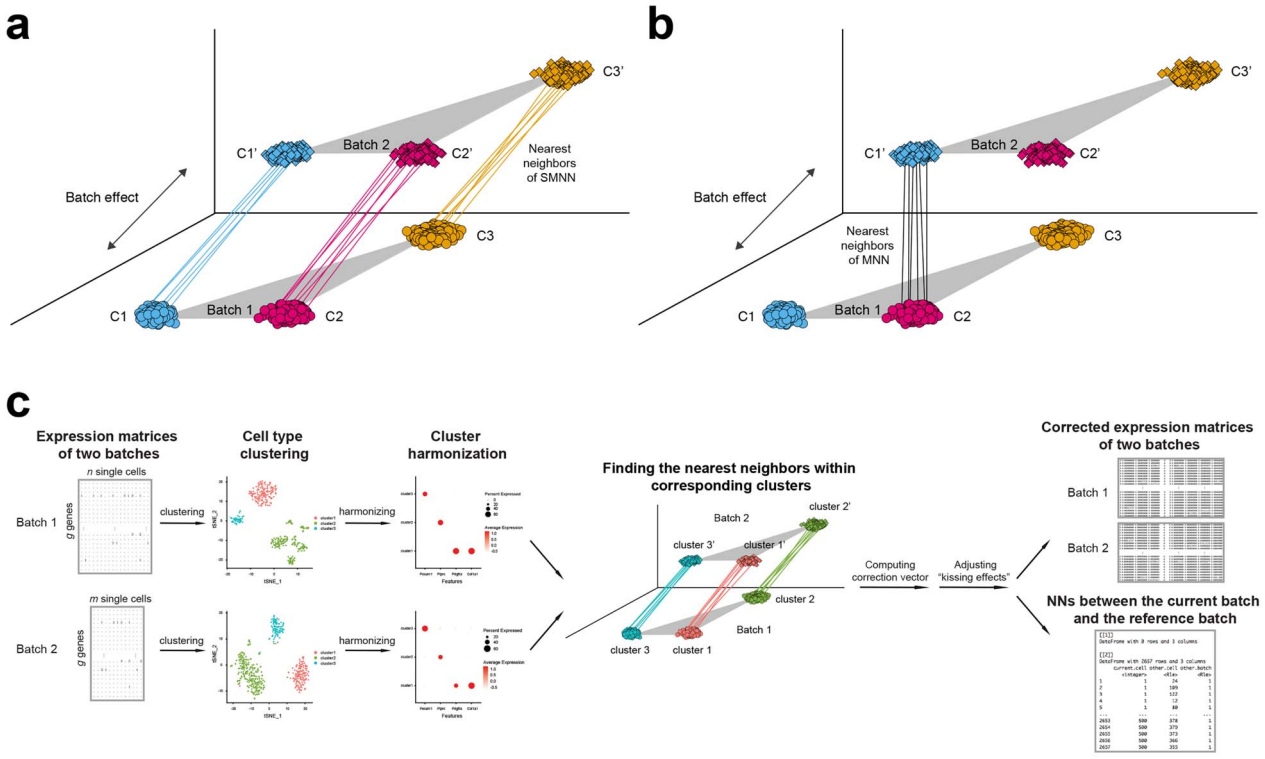

Figure 1. Overview of SMNN. Schematics for detecting MNNs between two batches under a non-orthogonal scenario (a) in SMNN and (b) in MNN. (c) Workflow of SMNN. Single cell clustering is first performed within each batch using Seurat v3; and then SMNN takes user-specified marker gene information for each cell type to match clusters/cell types across batches. With the clustering and cluster-specific marker gene information, SMNN searches MNNs within each cell type and performs batch effect correction accordingly.

across batches are brought in contact (rather than fully merged), commonly observed with naïve batch effect correction [8] (an example detailed in Supplementary Section 3 and visualized in Supplementary Figure S1). At the end of the second step, SMNN returns the batch-effect corrected expression matrix including all genes from the input matrix for each batch, as well as the information regarding nearest neighbors between the reference batch and the current batch under correction. This step is carried out for every batch other than the reference batch so that all batches are corrected to the same reference batch in the end.

Simulation results

Since MNN has been shown to excel alternative methods [4, 8], we here focus on comparing our SMNN with MNN. We first compared the performance of SMNN to MNN in simulated data. In our simulations, SMNN demonstrates superior performance over MNN under both orthogonal and non-orthogonal scenarios (Figures 2 and 3 and Supplementary Figures S2–S4). We show t-SNE plot for each cell type before and after MNN and SMNN correction under both the orthogonal and non-orthogonal scenarios. Under orthogonality, the two batches partially overlapped in the t-SNE plot before correction, suggesting that the variation due to batch effect was indeed much smaller than that due to biological effect. Both MNN and SMNN successfully mixed single cells from two batches (Supplementary Figure S3). However, for cell types 1 and 3, there were still some cells from the second batch left unmixed with those from the first batch after MNN correction (Supplementary Figure S3a and c). Under the non-orthogonal scenario, the differences between two batches were more pronounced before correction, and SMNN apparently outperformed MNN (Supplementary Figure S4), especially in cell type 1 (Supplementary Figure S4a). Moreover, we also computed Frobenius norm distance [20] for each cell between its simulated true profile before introducing batch effects and after SMNN and MNN correction. The results showed an apparently reduced deviation from the truth after SMNN correction than MNN (Figure 3). We have also simulated data using the original simulation framework in Haghverdi *et al.* [8], which does not allow precise control of orthogonality (detailed in Materials and Method section) and seems to simulate data closer to those under orthogonal cases (Supplementary Figure S5a). Applying SMNN and MNN to such simualted data, we also found that SMNN showed slight advantages (Supplementary Figure S5b). These results suggest that SMNN provides improved batch effect correction over MNN under both orthogonal and non-orthogonal scenarios.

Real data results

For performance evaluation in real data, we first carried out batch effect correction on two hematopoietic datasets (Supplementary Table S1) using four methods: our SMNN, published MNN, Seurat v3 and LIGER. Figure 4a–e shows UMAP plot before and after correction. Notably, all four methods can substantially mitigate discrepancy between the two datasets. Comparatively, SMNN better mixed cells of the same cell type across batches than the other three methods and seemed to better position cells from batch-specific cell types with respect to other biological-related cell types (Supplementary Figure S6

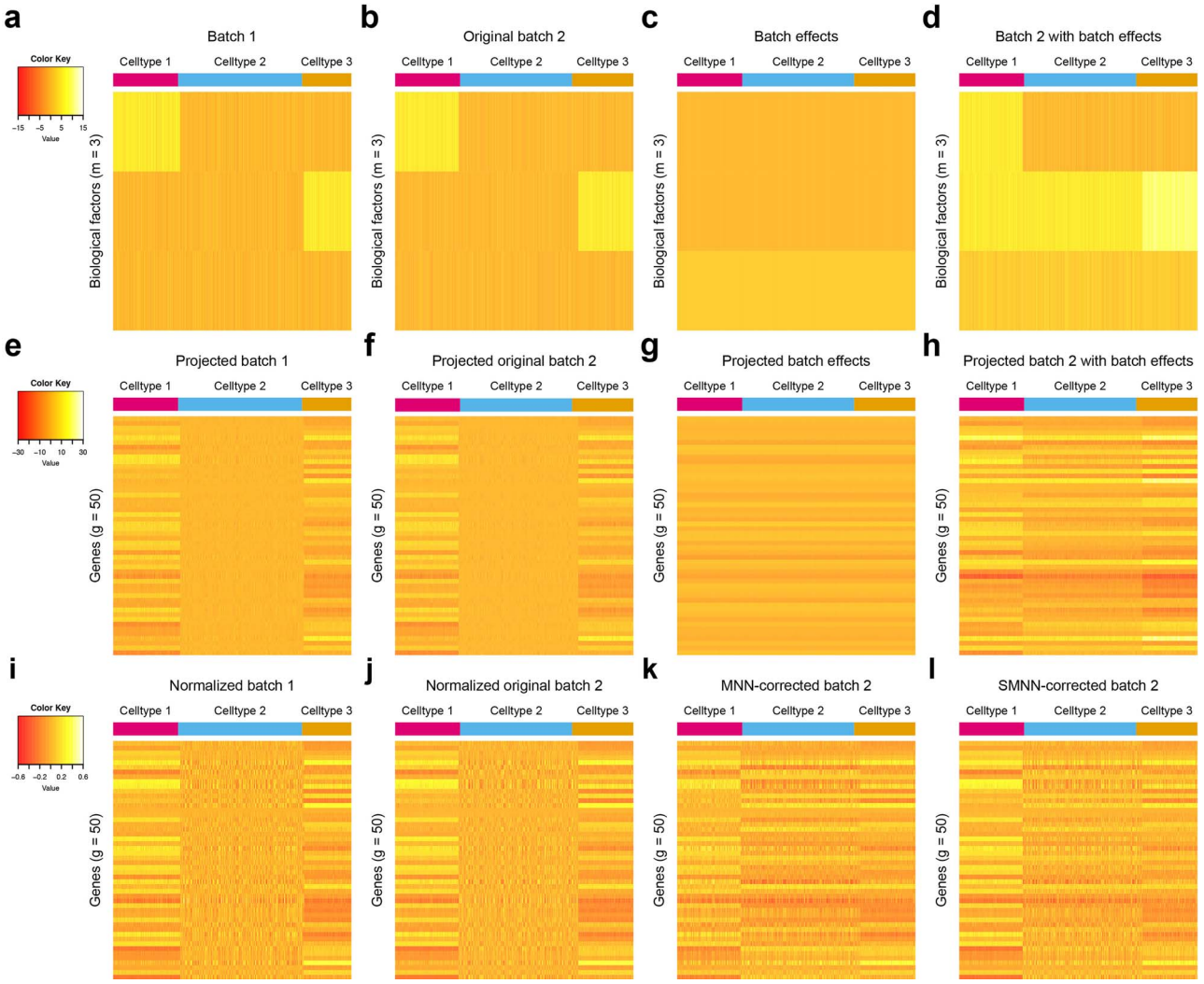

Figure 2. Heatmap of gene expression matrices for simulated data under non-orthogonal scenario. (a–d) 3D biological space with rows of each heatmap representing biological factors and columns corresponding to single cells. (e–h) High dimensional gene expression profiles with rows corresponding to genes and columns again representing single cells. (a, e and i) Correspond to the batch 1 and (b, f and j) correspond to batch 2. (c and g) Provide a visualization for the direction of batch effects in low-dimension biological space and high-dimension gene expression spaces, respectively. (d and h) Sum of (b) and (c) and sum of (f) and (g), respectively, are 'observed' data for cells in batch 2 in low and high dimensional space. (i and j) Are the cosine-normalized data for batch 1 and original batch 2. Note 'original' is in the sense that no batch effects have been introduced to the data yet. (k and l) Are the MNN and SMNN corrected results, respectively.

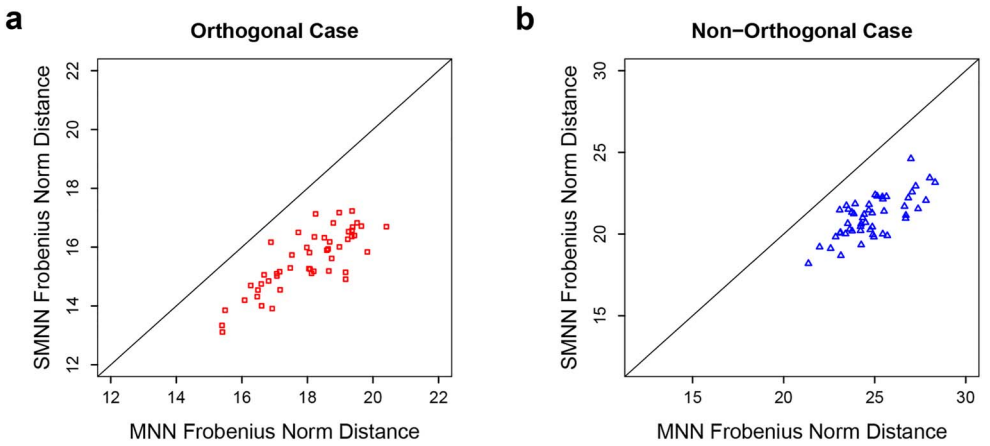

Figure 3. Frobenius norm distance between two batches after SMNN and MNN correction in simulation data under orthogonal (left) and non-orthogonal scenarios (right).

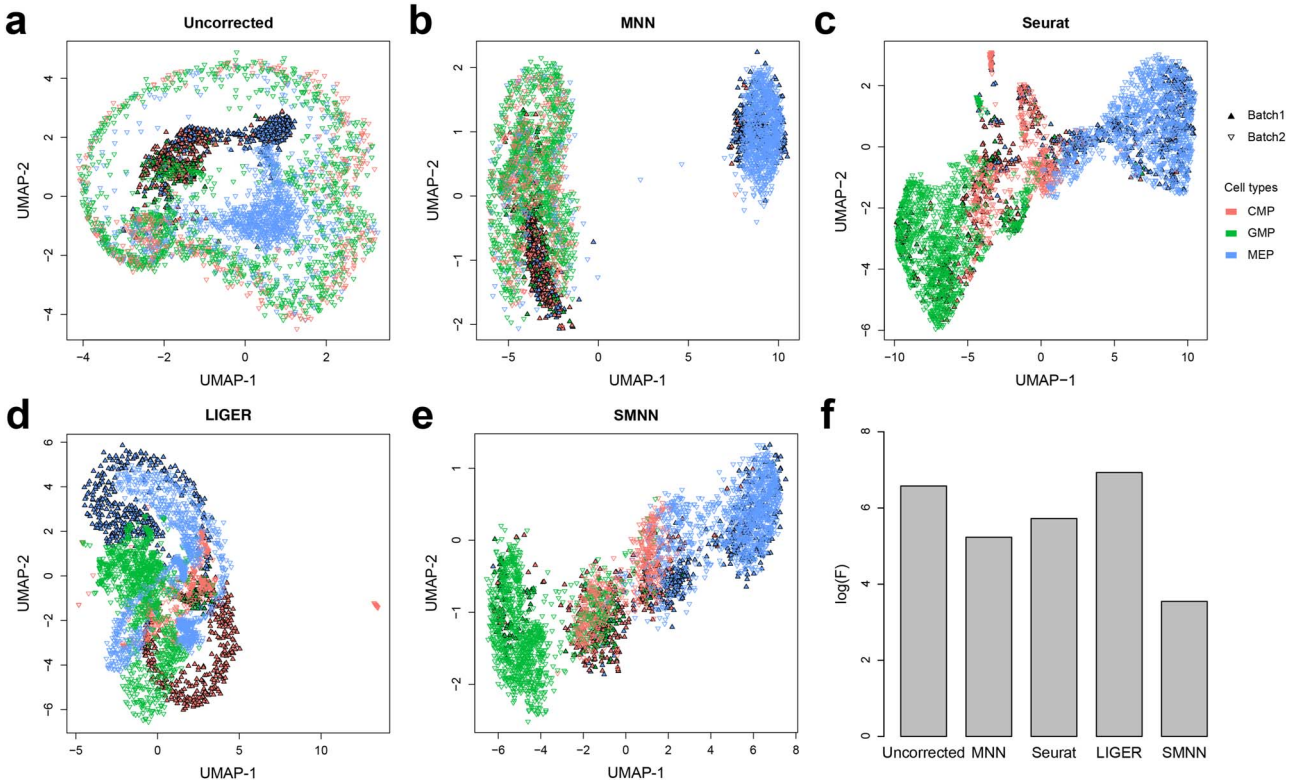

Figure 4. Performance comparison between SMNN and MNN in two hematopoietic datasets. (a) UMAP plots for two hematopoietic datasets before batch effect correction. Solid and inverted triangle represent the first and second batch, respectively; and different cell types are shown in different colors. (b–e) UMAP plots for the two hematopoietic datasets after correction with MNN, Seurat v3, LIGER and SMNN. (f) Logarithms of F-statistics for merged data of the two batches.

and S7), especially for common myeloid progenitor (CMP) and megakaryocyte-erythrocyte progenitor (MEP) cells, which were wrongly corrected by MNN due to sub-optimal nearest neighbor search ignoring cell-type information (Supplementary Figure S8). Correspondingly, SMNN corrected data exhibits the lowest *F* value than that from the other three methods. Specifically, *F* value is with reduced by 81.5–96.6% on top of MNN, Seurat v3 and LIGER, respectively (Figure 4f). Furthermore, we compared the distance for the cells between batch 1 and 2 and found that, compared with data before correction, both MNN and SMNN reduced the Euclidean distance between the two batches (Supplementary Figure S9). In addition, SMNN further decreased the distance by up to 8.2% than MNN [2.8%, 4.3% and 8.2% for cells of type CMP, MEP and granulocyte-monocyte progenitor (GMP) cells, respectively]. Under scenarios where we only have partial cell-type information, SMNN still better mixed cells of the same cell type across batches (detailed in Supplementary Section 3; Supplementary Figure S10a–c and e–g) and manifested the best/lowest *F* values, compared with uncorrected and MNN-corrected data (Supplementary Figure S10d and h). These results suggest improved batch effect correction by SMNN, compared with unsupervised correction methods.

SMNN identifies differentially expressed genes that are biologically relevant

We then compared the differentially expressed genes (DEGs) among different cell types identified by SMNN and MNN. After correction, in the merged hematopoietic dataset, 1012 and 1145 up-regulated DEGs were identified in CMP cells by SMNN and MNN, respectively, when compared with GMP cells, while 1126 and 1108 down-regulated DEGs were identified by the two methods, respectively (Figure 5a and Supplementary Figure S11a). Of them, 736 up-regulated and 842 down-regulated DEGs were shared between SMNN and MNN corrected data. Gene ontology (GO) enrichment analysis showed that the DEGs detected only by SMNN were overrepresented in GO terms related to blood coagulation and hemostasis, such as platelet activation and aggregation, hemostasis, coagulation and regulation of wound healing (Figure 5b). Similar DEG detection was carried out to detect genes differentially expressed between CMP and MEP cells. About 181 SMNN-specific DEGs were identified out of the 594 up-regulated DEGs in CMP cells when compared with MEP cells (Figure 5c), and they were found to be enriched for GO terms involved in immune cell proliferation and differentiation, including regulation of leukocyte proliferation, differentiation and migration, myeloid cell differentiation and mononuclear cell proliferation (Figure 5d). Lastly, genes identified by SMNN to be up-regulated in GMP when compared with MEP cells were found to be involved in immune processes, whereas up-regulated genes in MEP over GMP were enriched in blood coagulation (Supplementary Figure S11e–h). Comparatively, the GO terms enriched for MNN-specific DEGs seem not particularly relevant to corresponding cell functions (Supplementary Figure S12). These cell-function-relevant SMNN-specific DEGs indicate that SMNN can maintain some cell features that are missed by MNN after correction.

In addition, we considered two sets of 'working truth': first, DEGs identified in uncorrected batch 1 and, second, DEGs identified in batch 2, and we compared SMNN and MNN results to both sets of working truth. The results showed that, in both comparisons (one comparison for each set of working truth), fewer DEGs were observed in SMNN-corrected batch 2, but higher

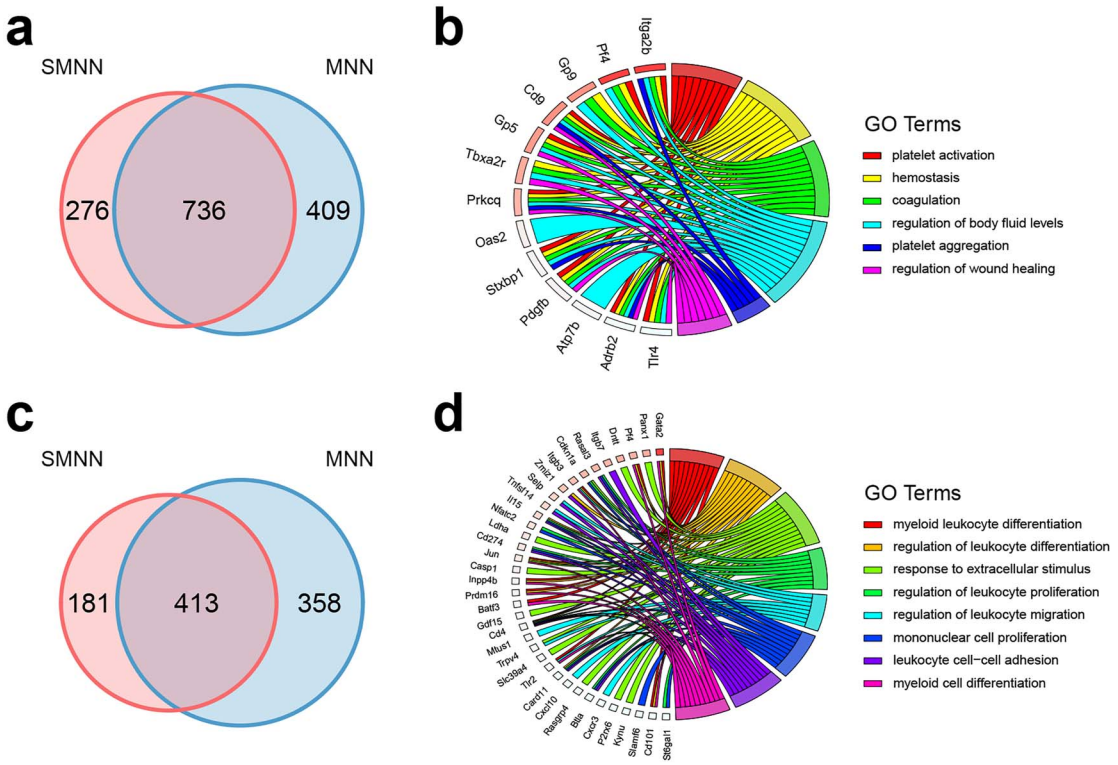

Figure 5. Comparison of DEGs, identified in the merged dataset by pooling batch 1 data with batch 2 data after SMNN and MNN correction. (a) Overlap of DEGs up-regulated in CMP over GMP after SMNN and MNN correction. (b) Feature-enriched GO terms and the corresponding DEGs up-regulated in CMP over GMP. (c) Overlap of DEGs up-regulated in CMP over MEP after SMNN and MNN correction. (d) Feature-enriched GO terms and the corresponding DEGs up-regulated in CMP over MEP.

precision and lower false negative rate in each of the three cell types than those in MNN results (Figure 6 and Supplementary Figures S13–S15). When compared with the uncorrected batch 1, 3.6–841.0% improvements in precision were observed in SMNN results than MNN (Figure 6 and Supplementary Figure S14). Similarly, SMNN increased the precision by 6.2–54.0% on top of MNN when compared with uncorrected batch 2 (Supplementary Figure S15). We also performed DEG analysis at various adjusted *P*-value thresholds, and the results showed that the better performance of SMNN is not sensitive to the *P*-value cutoff we used for DEG detection (detailed in Supplementary Section 3; Supplementary Figure S16). Such an improvement in the accuracy of DEG identification indicates that higher amount of information regarding cell structure was retained after SMNN correction than MNN.

We also identified DEGs between T cells and B cells in the merged human peripheral blood mononuclear cells (PBMCs) and T cell datasets after SMNN and MNN correction, respectively (Supplementary Figure S17). Compared with B cells, 3213 and 4180 up-regulated DEGs were identified in T cells by SMNN and MNN, respectively, 2203 of which were shared between the two methods (Supplementary Figure S17e). GO enrichment analysis showed that the SMNN-specific DEGs were significantly enriched for GO terms relevant to the processes of immune signal recognition and T cell activation, such as T cell receptor signaling pathway, innate immune response-activating signal transduction, cytoplasmic pattern recognition receptor signaling pathway and regulation of autophagy (Supplementary Figure S17f). In B cells, 5422 and 3462 were found to be up-regulated after SMNN and MNN correction, where 2765 were SMNN-specific (Supplementary Figure S17g). These genes were overrepresented in GO terms involved in protein synthesis and transport, including translational elongation and termination, ER to Golgi vesicle-mediated transport, vesicle organization and Golgi vesicle budding (Supplementary Figure S17h). These results again suggest that SMNN more accurately retains or rescues cell features after correction.

SMNN more accurately identifies cell clusters

Finally, we examined the ability to differentiate cell types after SMNN and MNN correction in three datasets (Supplementary Table S1). In all three real datasets, Adjusted Rand Index (ARI) after SMNN correction showed 7.6–42.3% improvements over that of MNN (Figure 7), suggesting that SMNN correction more effectively recovers cell-type specific features.

Discussion

In this study, we present SMNN, a batch effect correction method for scRNA-seq data via supervised MNN detection. Our work is built on the recently developed method MNN, which has showed advantages in batch effect correction than existing alternative methods. On top of MNN, our SMNN relaxes a strong assumption that underlies MNN: that the biological differentiations are orthogonal to batch effects [8]. When this fundamental assumption is violated, especially under the realistic scenario that the two batches are rather different, MNN tends to err when searching nearest neighbors for cells belonging to the same biological cell type across batches. Our SMNN, in contrast, explicitly considers cell-type label information to perform supervised MNN

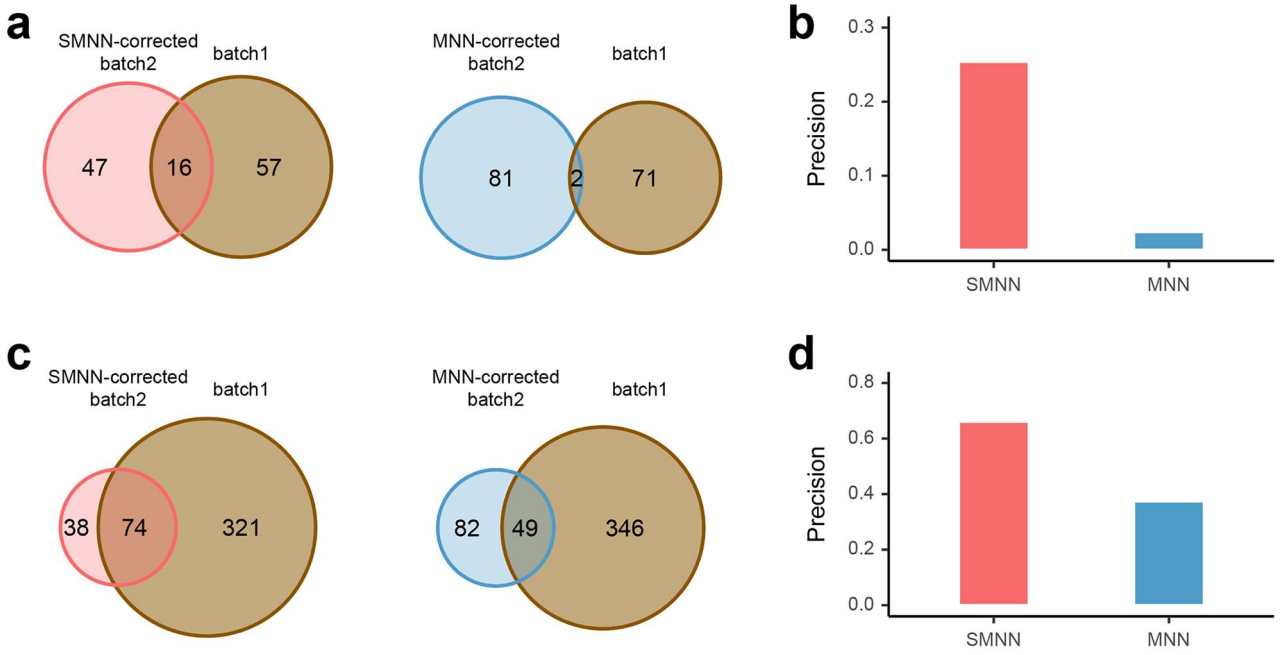

Figure 6. Reproducibility of DEGs (between CMP and GMP), identified in uncorrected batch 1 and in SMNN or MNN-corrected batch 2. (a) Reproducibility of DEGs up-regulated in CMP over GMP, detected in batch 1, versus SMNN (left) or MNN-corrected (right) batch 2. (b) TPR of the DEGs (between CMP and GMP) identified in batch 2 after SMNN and MNN correction. (c) Reproducibility of DEGs up-regulated in GMP over CMP, identified in the uncorrected batch 1, and in SMNN (left) or MNN-corrected (right) batch 2. (d) TPR of the DEGs up-regulated in GMP over CMP identified in batch 2 after SMNN and MNN correction.

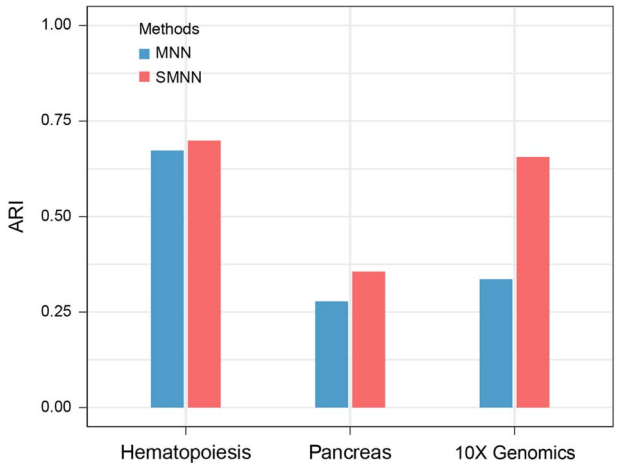

Figure 7. Clustering accuracy in three datasets after batch effect correction. ARI is employed to measure the similarity between clustering results before and after batch effect correction.

matching, thus empowered to extract only desired neighbors from the same cell type.

A notable feature of our SMNN is that it can detect and match the corresponding cell populations across batches with the help of feature markers provided by users. SMNN performs clustering within each batch before merging across batches, which can reveal basic data structure, i.e. cell composition and proportions of contributing cell types, without any adverse impact due to batch effects. Cells of each cluster are labeled by leveraging their average expression levels of certain marker(s), thus enabling us to limit the MNN detection within a smaller search space (i.e. only among cells of the same or similar cell type or status). This supervised approach eliminates the correction biases incurred by pairs of cells wrongly matched across cell types. We benchmarked SMNN together with three state-of-the-art batch effect correction methods, MNN, Seurat v3 and LIGER, on simulated and three published scRNA datasets. Our results clearly show the advantages of SMNN in terms removing batch effects. For example, our results for the hematopoietic datasets show that SMNN better mixed cells of all the three cell types across the two batches (Figure 4a–e) and reduced the differentiation between the two batches by up to 96.6% on top of the corrected results from the three unsupervised methods (Figure 4f), demonstrating that our SMNN method can more effectively mitigate batch effect. Additionally, cell population composition can also be a critical factor in batch effect correction. Our results by analyzing batches with varying cell type compositions (detailed in Supplementary Section 3; Supplementary Figure S18) suggest that our SMNN is robust to differential cell composition across batches.

More importantly, the wrongly matched cell pairs may wipe out the distinguishing features of cell types. This is mainly because, for a pair of cells from two different cell types, the true biological differentiations between them would be considered as technical biases and subsequently removed in the correction process. Compared with MNN, SMNN also appears to more accurately recover cell-type specific features: clustering accuracy using SMNN-corrected data increases substantially in all the three real datasets (by 7.6–42.3% when measured by ARI) (Figure 7). Furthermore, we observe power enhancement in detecting DEGs between different cell types in the data after SMNN correction than MNN (Figures 5 and 6 and Supplementary Figures S11–S15). Specifically, the precision of the DEGs identified by SMNN were improved by up to 841.0% and 54.0% than those by MNN when compared with the two set of working truth, respectively (Figure 6c and d and Supplementary Figures S14 and S15). Moreover, GO term enrichment results show that

the up-regulated DEGs identified only in SMNN-corrected GMP and MEP cells were involved in immune process and blood coagulation, respectively (Supplementary Figure S11f and h), which accurately reflect the major features of these two cell types [21]. Similarly, DEGs identified between T and B cells after SMNN correction are also biologically more relevant than those identified after MNN correction (Supplementary Figure S17f and h). These results suggest that SMNN can eliminate the overcorrection between different cell types and thus maintains more biological features in corrected data than MNN. Efficient removal of batch effects at reduced cost of biological information loss, manifested by SMNN in our extensive simulated and real data evaluations, empowers valid and more powerful downstream analysis.

In summary, extensive simulation and real data benchmarking suggest that our SMNN can not only better rescue biological features and thereof provide improved cluster results but also facilitate the identification of biologically relevant DEGs. Therefore, we anticipate that our SMNN is valuable for integrated analysis of multiple scRNA-seq datasets, accelerating genetic studies involving single-cell dynamics.

Materials and methods

Simulation framework

We simulated two scenarios, orthogonal and non-orthogonal, to compare the performance of MNN and SMNN. The difference between the two scenarios lies in the directions of the true underlying batch effect vectors with respect to those of the biological effects.

Baseline simulation

Our baseline simulation framework, similar to that adopted in Haghverdi *et al.* [8], contains two steps:

First, data are initially generated in low (specifically three) dimensional biological space. Data in each batch are independently generated from a Gaussian mixture model to represent a low dimensional biological space, with each component in the mixture corresponding to one cell type. Equations (1) and (2) below show formulae to generate two batches of such initial data, represented by matrices sets of vectors $\{X_k : k = 1, ..., n_1\}$ and $\{Y_l : l = 1, ..., n_2\}$, in low dimensional biological space.

$$X_k \sim \sum_{i=1}^{3} w_{1i} N(\mu_{1i}, I_3), with \sum_{i=1}^{3} w_{1i} = 1, and\ w_{11}, w_{12}, w_{13} \geq 0, \quad for\ k = 1, 2, \ldots, n_1 \tag{1}$$

$$Y_l \sim \sum_{j=1}^{3} w_{2j} N(\mu_{2j}, I_3), with \sum_{j=1}^{3} w_{2j} = 1, and\ w_{21}, w_{22}, w_{23} \geq 0, \quad for\ l = 1, 2, \ldots, n_2, \tag{2}$$

where μ_{1i} is the three-dimensional vector specifying cell-type specific means for the ith cell type in the first batch, reflecting the biological effect; similarly for μ_{2j}; n_1 and n_2 are the total number of cells in the first and second batch, respectively; w_{1i} and w_{2j} are the different mixing coefficients for the three cell types in the two batches and I_3 is the three-dimensional identity matrix with diagonal entries as ones and the rest entries as zeros. In our simulations, we set $n_1 = 1000$, $n_2 = 1100$ and

$$(w_{11}, w_{12}, w_{13}) = (0.3, 0.5, 0.2) \tag{3}$$

$$(w_{21}, w_{22}, w_{23}) = (0.25, 0.5, 0.25). \tag{4}$$

Secondly, we project the low dimensional data with batch effect to the high dimensional gene expression space. We map both datasets to $G = 50$ dimensions by linear transformation using the same random Gaussian matrix P, to simulate high-dimensional gene expression profiles.

$$\tilde{X_k} = PX_k, \text{for } k = 1, 2, \ldots, n_1 \tag{5}$$

$$\tilde{Y_l} = PY_l, \text{for } l = 1, 2, \ldots, n_2. \tag{6}$$

Here, P is a $G \times 3$ Gaussian random matrix with each entry simulated from the standard normal distribution.

Introduction of batch effects.

In Haghverdi *et al.* [8], batch effects are directly introduced in the high dimensional gene expression space. Specifically, a Gaussian random vector $b = \left(b_1, b_2, \ldots, b_G\right)^T$ is simulated and added to the second dataset via the following:

$$X_{Observed,k} = \tilde{X_k} + \varepsilon_{1,k}, for\ k = 1, 2, \ldots, n_1 \tag{7}$$

$$Y_{Observed,l} = \tilde{Y_l} + b + \varepsilon_{2,l}, for\ l = 1, 2, \ldots, n_2, \tag{8}$$

where $\tilde{X_k}$ and $\tilde{Y_l}$ are projected high-dimensional gene expression profiles; $\varepsilon_{1,k}$ and $\varepsilon_{2,l}$ are independent random noises added to the expression of each 'gene' for each cell in the two batches.

In our simulations, we adopt a different approach: we introduce batch effects in the low dimensional biological space. Specifically, we simulate a bias vector $c = \left(c_1, c_2, c_3\right)^T$ in the biological space

$$X_{Observed,k} = \tilde{X_k} + \varepsilon_{1,k} = PX_k + \varepsilon_{1,k}, for\ k = 1, 2, \ldots, n_1 \tag{9}$$

$$\begin{aligned} Y_{Observed,l} &= \tilde{Y_{SMNN,l}} + \varepsilon_{2,l} = P(Y_l + c) + \varepsilon_{2,l} \\ &= PY_l + Pc + \varepsilon_{2,l}, \quad for\ l = 1, 2, \ldots, n_2. \end{aligned} \tag{10}$$

Our simulation framework can be viewed as a reparametrized version of the model in Haghverdi *et al.* [8]. For each batch effect b of the model in Haghverdi *et al.* [8], there exist multiple pairs of projection matrix P and vector c such that $b = Pc$, and for any vector c in our model, there is a corresponding vector $b = Pc$ given a fixed projection matrix P. In particular,$(b)_l = \left(Pc\right)_l = \sum_{i=1}^{G} P_{li} c_i \sim N\left(0, \sum_{i=1}^{G} c_i^2\right)$. In other words, for any simulated setting in Haghverdi *et al.* [8], we can find at least one equivalent setting in our model, and vice versa. Although our simulation framework is largely similar to that in Haghverdi *et al.* [8], the two differ in the following two aspects:

First, the low dimensional biological space is three-dimensional in ours and two-dimensional in Haghverdi *et al.* [8].

Second, we introduce batch effects c in low dimensional biological space and then projected to high dimensional space (Equation 10), while Haghverdi *et al.* [8] directly introduce batch

effects b in the high dimensional gene expression space (Equation 8). We made such changes so that we can simulate both the orthogonal and non-orthogonal scenarios in a more straightforward manner that the extent of orthogonality can be controlled (equation 11). The orthogonality is defined in the sense that biological differences (that is, mean difference between any two clusters/cell-types) are orthogonal to those from batch effects.

Our framework allows flexible modelling of the biological effects and batch effects in the same low dimensional biological space and allows us to control the extent of orthogonality. Specifically, the batch effect c is added to mean vectors of three cell types in batch 1 to get the mean vectors of three cell types for batch 2.

$$\mu_{2i} = \mu_{1i} + c, for\ i = 1, 2, 3. \tag{11}$$

Note that $\left(\mu_{1j} - \mu_{1i}\right)c = 0, for\ i \neq j \in \left\{1, 2, 3\right\}$ represents the orthogonal scenario that variation from batch effect is orthogonal to mean difference between any two clusters/cell-types, and $\left(\mu_{1j} - \mu_{1i}\right)c \neq 0, for\ i \neq j \in \left\{1, 2, 3\right\}$ in the non-orthogonal case.

Leveraging the simulation framework described before, we simulate two scenarios via the following:

(i) In the orthogonal case, we set $c = \left(0, 0, 2\right)^T$

(a) $\mu_{11} = \left(5, 0, 0\right)^T, \mu_{12} = \left(0, 0, 0\right)^T, \mu_{13} = \left(0, 5, 0\right)^T$

(b) $\mu_{21} = \left(5, 0, 2\right)^T, \mu_{22} = \left(0, 0, 2\right)^T, \mu_{23} = \left(0, 5, 2\right)^T$

(ii) In the non-orthogonal case, we set $c = \left(0, 5, 2\right)^T$

(a) $\mu_{11} = \left(5, 0, 0\right)^T, \mu_{12} = \left(0, 0, 0\right)^T, \mu_{13} = \left(0, 5, 0\right)^T$

(b) $\mu_{21} = \left(5, 5, 2\right)^T, \mu_{22} = \left(0, 5, 2\right)^T, \mu_{23} = \left(0, 10, 2\right)^T$

Performance evaluation

MNN and SMNN share the goal to correct batch effects. Mathematically, using the notations introduced in baseline simulation, the goal translates into de-biasing vector c (which would be effectively reduced to b in the orthogonal case). Without loss of generality and following MNN, we treat the first batch as the reference and correct the second batch $\left\{Y_{\text{Observed},l} : l = 1, \ldots, n_2\right\}$ to the first batch $\left\{X_{\text{Observed},k} : k = 1, \ldots, n_1\right\}$. Denote the corrected values from MNN and SMNN as $\left\{\hat{Y}_{\text{MNN},l} : l = 1, \ldots, n_2\right\}$ and $\left\{\hat{Y}_{\text{SMNN},l} : l = 1, \ldots, n_2\right\}$, respectively.

To measure the performance of the two correction methods, we utilize the Frobenius norm [20] to define the loss function

$$L\left(\hat{Y}, \tilde{Y}\right) = \left\|\tilde{Y} - \hat{Y}\right\|_F = \sqrt{\sum_{l=1}^{n_2}\left\|\tilde{Y}_l - \hat{Y}_l\right\|^2} = \sqrt{\sum_{l=1}^{n_2}\sum_{g=1}^{G}\left|\tilde{Y}_{l,g} - \hat{Y}_{l,g}\right|^2}, \tag{12}$$

where $\tilde{Y} = \left[\tilde{Y}_1, \ldots, \tilde{Y}_k, \ldots, \tilde{Y}_{n_2}\right], \hat{Y} = \left[\hat{Y}_1, \ldots, \hat{Y}_k, \ldots, \hat{Y}_{n_2}\right]$. Note that $\tilde{Y}$ is the simulated true profiles introduced in Equations (5) and (6) before batch effects, and noises are introduced in Equations (7) and (8). Since MNN conducts cosine normalization to the input and the output, we use cosine-normalized $\tilde{Y}$ when calculating the above loss function.

Real data benchmarking

To assess the performance of SMNN in real data, we compared SMNN to alternative batch effect correction methods: MNN [8], Seurat v3 [17] and LIGER [22] to two hematopoietic scRNA-seq datasets, generated using different sequencing platforms, MARS-seq and SMART-seq2 (Supplementary Table S1) [10, 23]. The first batch produced by MARS-seq consists of 1920 cells of six major cell types, and the second batch generated by SMART-seq2 contains 2730 of three cell types, where three cell types, CMP, GMP and MEP cells, are shared between these two batches (here the two datasets). Batch effect correction was carried out using all four methods, following their default instructions. Cell-type labels were fed to SMNN directly according to the annotation from the original papers. To better compare the performance between MNN and SMNN, only the three cell types shared between the two batches were extracted for our downstream analyses. The corrected results of all the three cell types together, as well as for each of them separately, were visualized by UMAP using *umap-learn* method [24]. In order to qualify the mixture of single cells using both batch correction methods, we calculated: (i) F statistics under two-way multivariate analysis of variance (MANOVA) for merged datasets of the two batches. F statistics quantifies differences between batches, where smaller values indicating better mixing of cells across batches and (ii) the distance for the cells within each cell type in batch 2 to the centroid of the corresponding cell group in batch 1.

To measure the separation of cell types after correction, we additionally attempted to detect DEGs between different cell types in both SMNN and MNN corrected datasets. The corrected expression matrices of the two batches were merged and DEGs were detected by Seurat v3 using Wilcoxon rank sum test [17]. Genes with an adjusted P-value <0.01 were considered as differentially expressed. GO enrichment analysis was performed for the DEGs exclusively identified by SMNN using *clusterProfiler* [25]. Because there is no ground truth for DEGs, we further identified DEGs between different cell types within corrected batch 2 and then compared them to those identified in uncorrected batch 1 and uncorrected batch 2, which supposedly are not affected by the choice of batch effect correction method. True positive rate (TPR) was computed for each comparison.

Additionally, we also performed batch effect correction on another two tissues/cell lines, pancreas [26, 27] and PBMCs [28], again using both SMNN and MNN. DEGs were detected between T cells and B cells in the merged PBMC and T cell datasets after SMNN and MNN correction, respectively. Furthermore, single cell clustering was applied to batch-effects corrected gene expression matrices in all the three real datasets following the pipeline described in Haghverdi *et al.* [8]. Cell-type labels before correction were considered as ground truth, and ARI [29] was employed to measure the clustering similarity before and after correction

$$\text{ARI}\left(L_q, L_s\right) = \frac{\sum_{q,s}\binom{n_{qs}}{2} - \left[\sum_q \binom{n_q}{2}\sum_s \binom{n_s}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_q \binom{n_q}{2} + \sum_s \binom{n_s}{2}\right] - \left[\sum_q \binom{n_q}{2}\sum_s \binom{n_s}{2}\right] \Big/ \binom{n}{2}}, \tag{13}$$

where n_q and n_s are the single cell numbers in cluster q and s, respectively; n_{qs} is the number of single cells shared between clusters q and s; and n is the total number of single cells. ARI

ranges from 0 to 1, where a higher value represents a higher level of similarity between the two sets of cluster labels.

Data and software availability

SMNN is compiled as an R package and freely available at https://yunliweb.its.unc.edu/SMNN/ and https://github.com/yycunc/SMNN. The data we adopted for benchmarking at from following: (i) two Mouse hematopoietic scRNA-seq datasets from [10] (GEO accession number GSE81682) and [23] (GEO accession number GSE72857); (ii) two human pancreas scRNA-seq datasets from [26] (GSE81076) and [27] (GSE85241) and (iii) two 10X Genomics datasets of PBMCs and T cells from [28] (https://support.10xgenomics.com/single-cell-gene-expression/datasets/).

Key Points

- Batch effect correction has been recognized to be critical when integrating scRNA-seq data from multiple batches due to systematic differences in time points, generating laboratory and/or handling technician(s), experimental protocol and/or sequencing platform.
- Existing batch effect correction methods that leverage information from mutual nearest neighbors (MNNs) across batches (for example, implemented in MNN or Seurat) ignores cell-type information and suffers from potentially mismatching single cells from different cell types across batches, which would lead to undesired correction results, especially under the scenario where variation from batch effects is non-negligible compared with biological effects.
- To address this critical issue, here, we present SMNN, a supervised machine learning method that first takes cluster/cell-type label information from users or inferred from scRNA-seq clustering, and then searches MNNs within each cell type instead of global searching.
- Our SMNN method shows clear advantages over three state-of-the-art batch effect correction methods and can better mix cells of the same cell type across batches and more effectively recover cell-type specific features, in both simulations and real datasets.

Author Contributions

Y.L. initiated and designed the study. Y.Y., G.L. and H.Q. implemented the model and performed simulation studies and benchmarking evaluation. Y.Y, G.L. and Y.L. wrote the manuscript, and all authors edited and revised the manuscript.

Supplementary Data

Supplementary data are available online at https://academic.oup.com/bib.

Funding

National Institute of Health grants [R01 HL129132 Y.L. and R01 GM105785].

Conflict of Interest

The authors declare no competing interests.

References

1. Rozenblatt-Rosen O, Stubbington MJ, Regev A, *et al*. The human cell atlas: from vision to reality. *Nat News* 2017;**550**:451.
2. Stegle O, Teichmann SA, Marioni JC. Computational and analytical challenges in single-cell transcriptomics. *Nat Rev Genet* 2015;**16**:133.
3. Chen M, Zhou X. Controlling for confounding effects in single cell RNA sequencing studies using both control and target genes. *Sci Rep* 2017;**7**:13587.
4. Stuart T, Satija R. Integrative single-cell analysis. *Nat Rev Genet* 2019;**20**:257–72.
5. Smyth GK. *Limma: linear models for microarray data. Bioinformatics and computational biology solutions using R and Bioconductor*. New York, NY: Springer, 2005, 397–420.
6. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**:118–27.
7. Leek JT. Svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res* 2014;**42**:e161.
8. Haghverdi L, Lun AT, Morgan MD, *et al*. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* 2018;**36**:421.
9. Van Der Maaten L. Accelerating t-SNE using tree-based algorithms. *J Mach Learn Res* 2014;**15**:3221–45.
10. Nestorowa S, Hamey FK, Sala BP, *et al*. A single-cell resolution map of mouse hematopoietic stem and progenitor cell differentiation. *Blood* 2016;**128**:e20–31.
11. Spitzer MH, Gherardini PF, Fragiadakis GK, *et al*. An interactive reference framework for modeling a dynamic immune system. *Science* 2015;**349**:1259425.
12. Stuart T, Butler A, Hoffman P, *et al*. Comprehensive integration of single-cell data. *Cell* 2019;**177**:1888–902.
13. Duò A, Robinson MD, Soneson C. A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Res* 2018;**7**:1141.
14. Kiselev VY, Andrews TS, Hemberg M. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat Rev Genet* 2019;**20**:273–82.
15. Zhu L, Lei J, Klei L, *et al*. Semisoft clustering of single-cell data. *P Natl Acad Sci USA* 2019;**116**:466–71.
16. Sun Z, Chen L, Xin H, *et al*. A Bayesian mixture model for clustering droplet-based single-cell transcriptomic data from population studies. *Nat Commun* 2019;**10**:1649.
17. Butler A, Hoffman P, Smibert P, *et al*. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* 2018;**36**:411–20.
18. Yang Y, Huh R, Culpepper HW, *et al*. SAFE-clustering: single-cell aggregated (from ensemble) clustering for single-cell RNA-seq data. *Bioinformatics* 2019;**35**:1269–77.
19. Huh R, Yang Y, Jiang Y, *et al*. SAME-clustering: single-cell aggregated clustering via mixture model ensemble. *Nucleic Acids Res* 2020;**48**:86–95.

20. Van Loan CF, Golub GH. *Matrix computations*. Baltimore: Johns Hopkins University Press, 1983.
21. Lieu YK, Reddy EP. Impaired adult myeloid progenitor CMP and GMP cell function in conditional c-myb-knockout mice. *Cell Cycle* 2012;**11**:3504–12.
22. Welch JD, Kozareva V, Ferreira A, *et al*. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell* 2019;**177**:1873–87.
23. Paul F, Ya A, Giladi A, *et al*. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell* 2015;**163**:1663–77.
24. Becht E, McInnes L, Healy J, *et al*. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat Biotechnol* 2019;**37**:38–44.
25. Yu G, Wang L-G, Han Y, *et al*. clusterProfiler: an R package for comparing biological themes among gene clusters. *OMICS* 2012;**16**:284–7.
26. Grün D, Muraro MJ, Boisset J-C, *et al*. De novo prediction of stem cell identity using single-cell transcriptome data. *Cell Stem Cell* 2016;**19**:266–77.
27. Muraro MJ, Dharmadhikari G, Grün D, *et al*. A single-cell transcriptome atlas of the human pancreas. *Cell Syst* 2016;**3**:385–94 e383.
28. Zheng GX, Terry JM, Belgrader P, *et al*. Massively parallel digital transcriptional profiling of single cells. *Nat Commun* 2017;**8**:14049.
29. Hubert L, Arabie P. Comparing partitions. *J Classif* 1985;**2**:193–218.