Batch Effect Correction of RNA-seq Data through Sample Distance Matrix Correction

Teng Fei[1]**, Tianwei Yu**[1]

[1]Department of Biostatistics and Bioinformatics, Emory University, Atlanta, GA

Contact: `teng.fei@emory.edu`

EMORY ROLLINS SCHOOL OF PUBLIC HEALTH | Department of Biostatistics and Bioinformatics

Motivation

Batch effects is a frequent challenge in second generation sequencing data analysis. Several batch effect removal algorithms have shown good sample clustering results, providing important sample pattern information in corrected distance matrices. However downstream analyses, such as differential expression analysis, still suffer from remaining artifacts.

We have previously developed a method (Fei *et al*, 2018) to correct the sample distance matrix, resulting in good sample clustering results. However the method doesn't correct the original count data matrix. In the current work, utilizing the corrected distance matrix as reference distance matrix, we numerically solve a novel least squares loss function to conduct linear transformation on the raw count matrix. The resulting corrected count matrix yields Pearson correlation that approximates the sample pattern reflected by the reference distance matrix. Downstream analyses can then be performed on the corrected count matrix.

Availability: The R package is available at: **https://github.com/tengfei-emory/scBatch**

Problem setup

- X: $p \times n$ count matrix subject to batch effects.
- D_0: $n \times n$ distance matrix output from correction method `QuantNorm` (Fei *et al*, 2018).
- W: $n \times n$ weight matrix to be optimized, such that the Pearson correlation of the linear-transformed count matrix $Y = XW$ approximates the sample pattern in D_0.
- $Y = XW$: $p \times n$ corrected count matrix used in downstream analyses.
- D_Y: the Pearson correlation matrix of Y.
- $||\cdot||_F$: Frobenius norm.
- Define $C = X^T(I_p - \frac{1}{p}\mathbb{1}_p\mathbb{1}_p^T)^2X$.

Loss function and algorithm

In order to solve W, we propose to minimize the following least squares loss function

$$L(W) = \frac{1}{2}||D_Y - D_0||_F^2 = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}|D_{Yij} - D_{0ij}|^2,$$

Thus, the optimized weight matrix W_{opt} satisfies $W_{opt} = \text{argmin}_W L(W)$ and the corrected count matrix is $Y_{opt} = XW_{opt}$. By chain rule, the gradient of the loss function $L(W)$ is

$$\frac{\partial}{\partial W}L(W) = \left(\frac{\partial}{\partial W}D_Y\right)^T(D_Y - D_0).$$

By some algebra, the columnwise gradient satisfies

$$\begin{aligned}\frac{\partial}{\partial W_k}L(W) =& (\frac{\partial}{\partial W_k}D_Y)^T\{D_{Yk} - D_k\} \\ &+ \text{trace}\left[\left(\frac{\partial}{\partial W_k}D_Y\right)^T\{D_Y - D\}\right]\mathbf{e_k},\end{aligned}$$

where $\mathbf{e_k}$ is a $n \times 1$ vector with kth entry equal to one and others equal to zero and

$$\begin{aligned}\frac{\partial}{\partial W_k}D_Y =& \left[\frac{CW}{(W_k^TCW_k)^{1/2}} - \frac{CW_k(W^TCW_k)^T}{(W_k^TCW_k)^{3/2}}\right] \\ &\odot\left[\{\mathbb{1}_n \otimes \text{diag}(W^TCW)^{\circ 1/2}\}^{\circ -1}\right],\end{aligned}$$

where $\odot, \circ$ respectively represents Hadamard product and power, $\otimes$ represents outer product.

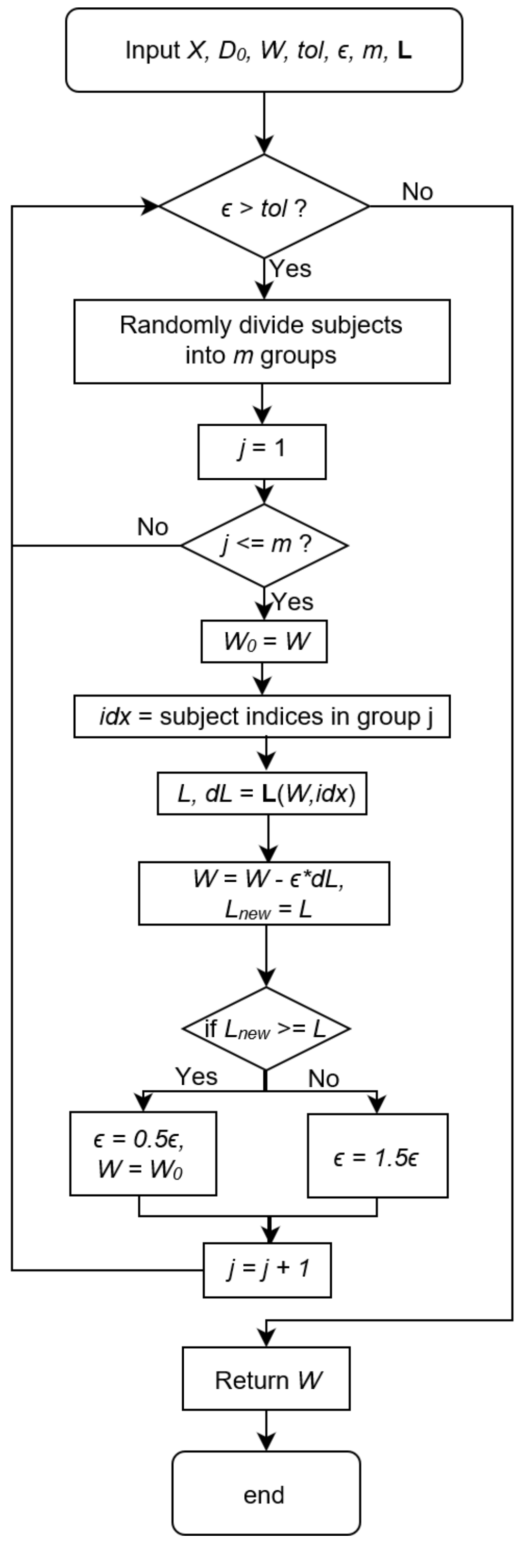

Figure 1: Algorithm flow chart.

We adapt a flexible gradient descent algorithm (Figure 1) which updates m randomly chosen columns in W in each iteration. The flexibility alleviated both the long running time of coordinate gradient descent and the underperformed result of full gradient descent algorithm. In order to dynamically adjust the learning rate, we utilized Armijo line search. The algorithm is stopped when the step size ϵ is less than a threshold, indicating approximation to a local minimum.

Simulation study

We applied Bioconductor package `splatter` (Zappia *et al*, 2017) to simulate single-cell RNA-seq data. Four batches with equal size were simulated, each containing the same four biological groups with different proportions. The probability of each gene to be a DE gene was set to 0.1. The simulated data sets contained DE factors for each gene in each group, providing a gold standard to assess DE gene detection. `edgeR` (Robinson *et al*, 2010) was used for DE gene detection.

We compared the DE gene detection performance with two popular batch effect removal methods: `ComBat` (Johnson *et al*, 2007) and `mnnCorrect` (Haghverdi *et al*, 2018). As can be observed in Figure 2, the proposed method consistently achieved better area under the ROC curve (AUC) and area under the precision-recall curve (PRAUC), indicating a better DE gene detection.

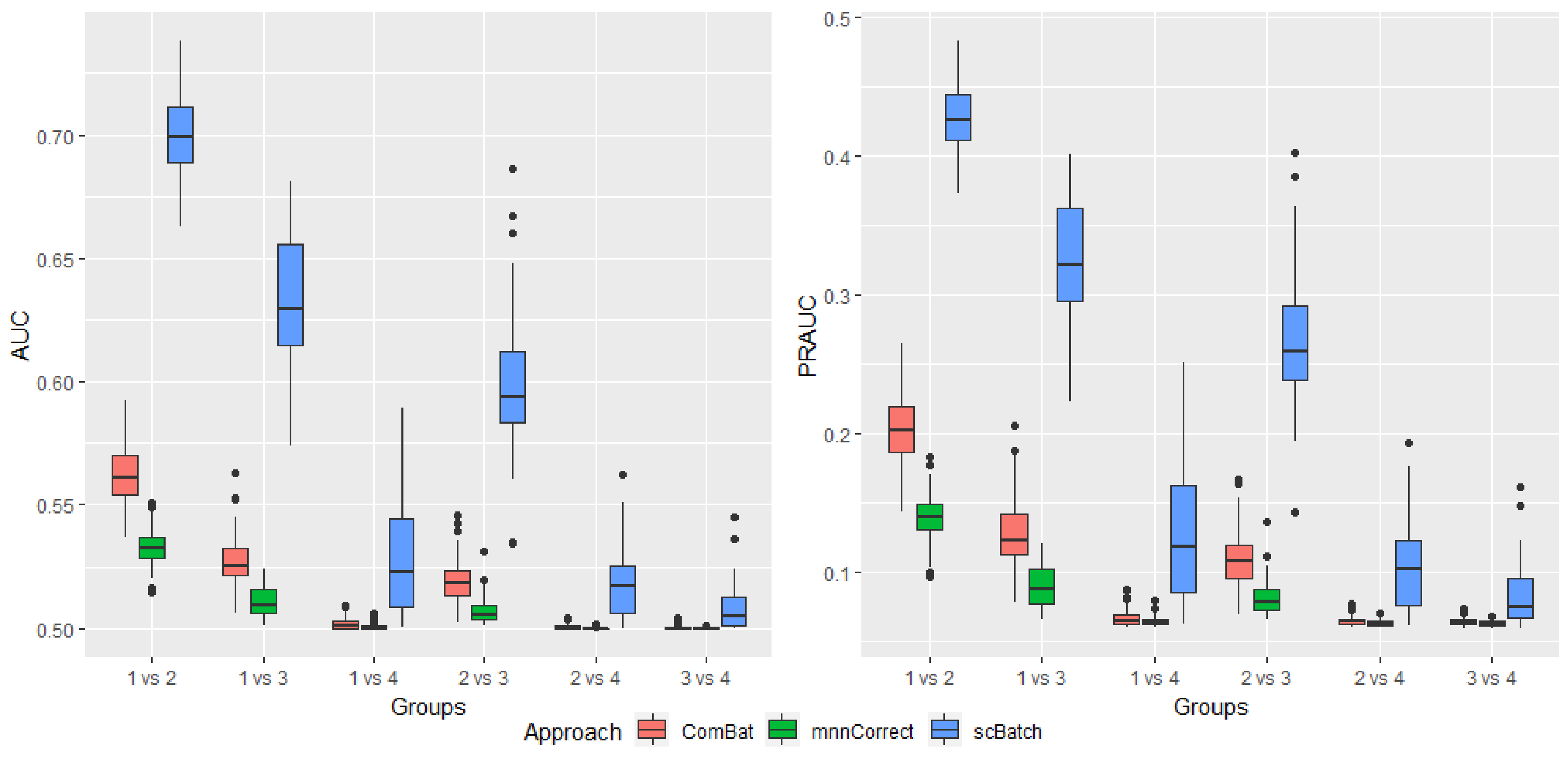

Figure 2: The DE gene detection results of 50 simulations when sample size is 200 and number of genes is 25000. The composition of the four biological groups is 40%, 30%, 15% and 15% respectively.

ENCODE Human and mouse tissues data

For real data analysis, we first conducted clustering analysis on the ENCODE Human and mouse tissues data (Lin *et al*, 2014) using the three methods. As Figure 3 displays, our method retrieved most biological pattern by matching most pairs of tissues. The other two methods also corrected batch effects to some extent.

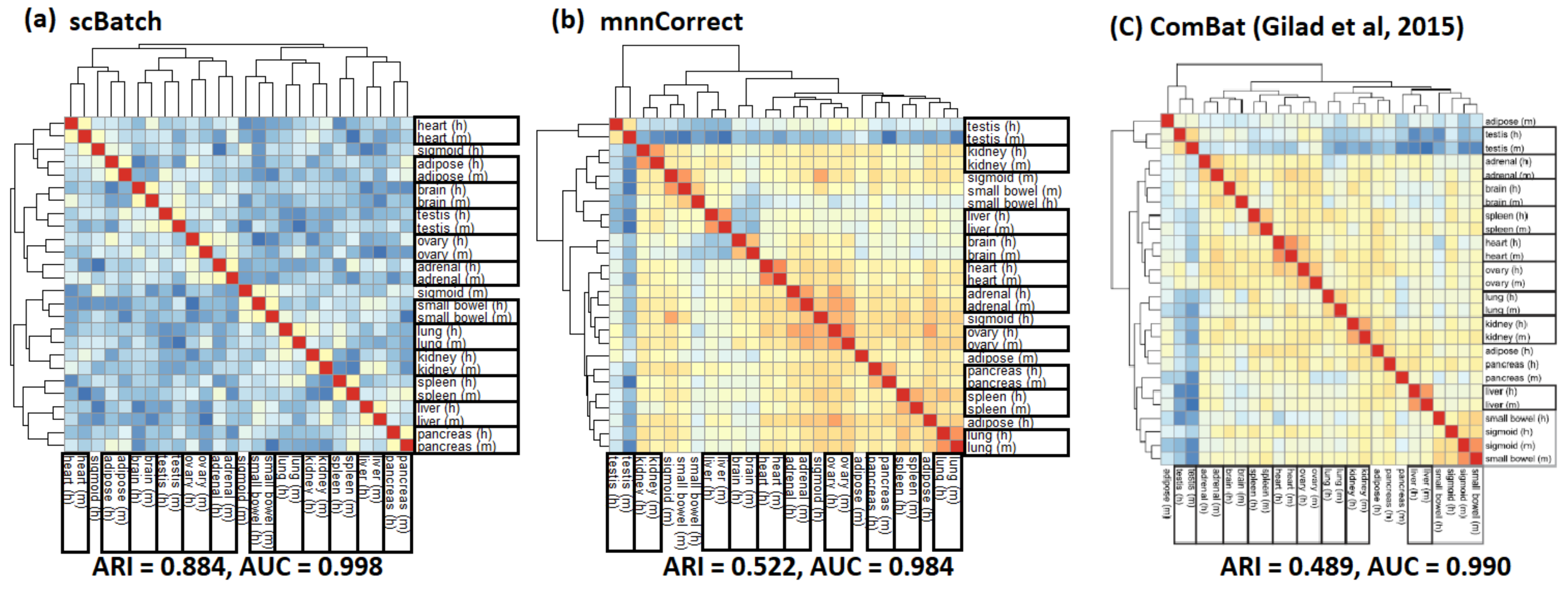

Figure 3: Heatmap for the corrected ENCODE data by three approaches.

Mouse neuron scRNA-seq data GSE59739

The GSE59739 scRNA-seq data (Usoskin *et al*, 2015) contains 25334 genes and 622 cells. As illustrated by Fei *et al*, 2018, although the biological pattern was largely retained in the raw data, there were still batch effects between samples from different libraries. Thus, we conducted batch effect correction for the data so as to investigate whether clustering and DE gene analysis would be affected by batch effects. As Figure 4 and Figure 5 demonstrate, our method outperforms the other two methods in both clustering and DE gene detection.

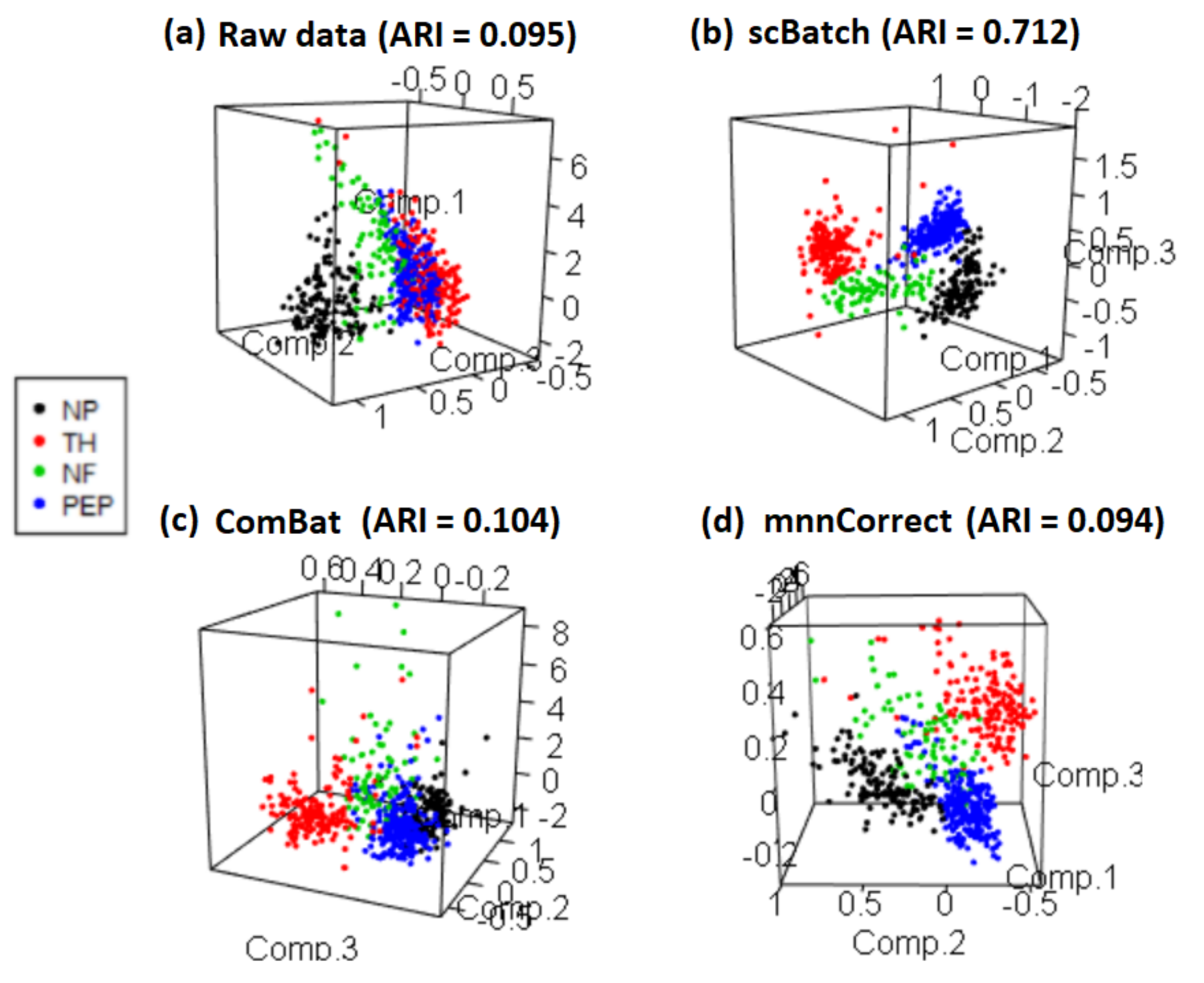

Figure 4: 3D PCA plots for raw and corrected data by three approaches. Four different colors refer to the four cell types. As can be seen, the batch effect is weak in the raw data, while the proposed method can still improve the sample pattern.

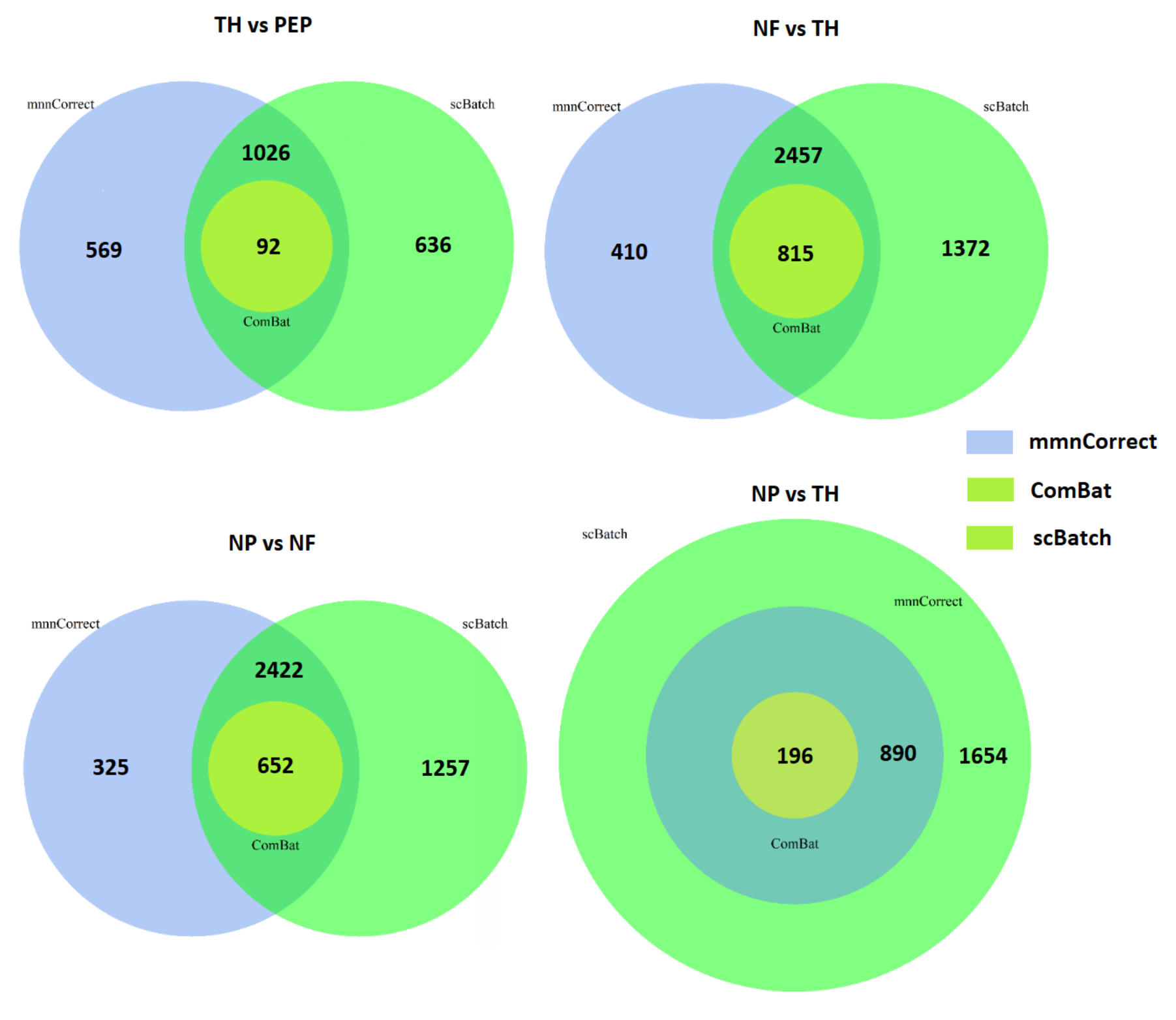

Figure 5: Venn diagram for detected DE genes (by `edgeR`, q-value < 0.05) using corrected data from three approaches for selected four pairs of cell types. The proposed method helps detect more DE genes, especially for cell types tangled before correction.

References

Fei, Teng, et al. ”Mitigating the adverse impact of batch effects in sample pattern detection.” Bioinformatics 34.15 (2018), 2634-2641.

Gilad, Yoav, and Orna Mizrahi-Man. ”A reanalysis of mouse ENCODE comparative gene expression data.” *F1000Research* 4 (2015).

Haghverdi, Laleh, et al. ”Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors.” *Nature biotechnology* 36.5 (2018): 421.

Johnson,W.E. et al. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8, (2007) 118-127.

Lin, Shin, et al. ”Comparison of the transcriptional landscapes between human and mouse tissues.” *Proceedings of the National Academy of Sciences* 111.48 (2014): 17224-17229.

Robinson, Mark D. et al. ”edgeR: a Bioconductor package for differential expression analysis of digital gene expression data.” *Bioinformatics* 26.1 (2010): 139-140.

Usoskin, Dmitry, et al. ”Unbiased classification of sensory neuron types by large-scale single-cell RNA sequencing.” *Nature neuroscience* 18.1 (2015): 145.

Zappia, Luke, et al ”Splatter: simulation of single-cell RNA sequencing data.” *Genome biology* 18.1 (2017): 174.

Acknowledgements

This work was partially funded by NIH grants R01GM124061, R37AI051231, U19AI057266.