Deep learning based multi-batch calibration for classification in various omics

Qian Wang (wang.qian@sjtu.edu.cn)
Shanghai Jiao Tong University
Jingyang Niu
Shanghai Jiao Tong University
Wei Xu
Shanghai Jiao Tong University
Dongming Wei
Shanghai JiaoTong University
Kun Qian
Shanghai Jiao Tong University https://orcid.org/0000-0003-1666-1965

Article

Keywords:

Posted Date: December 2nd, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1017151/v1

Deep learning based multi-batch calibration for classification in various omics

JingYang Niu[1,#], Wei Xu, DongMing Wei, Kun Qian, Qian Wang[1,*]

Abstract

Background: The amount of available biological data has exploded since the emergence of high-throughput technologies, which is not only revolting the way we recognize molecules and diseases but also bringing novel analytical challenges to bioinformatics analysis. In the last decade, deep learning has become a dominant technique in data science. However, classification accuracy is plagued with domain discrepancy. Notably, in the presence of multiple batches, domain discrepancy typically happens between individual batches. The recently proposed pair-wise adaptation approach may be suboptimal as it fails to eliminate the external factors across multiple batches and takes the classification task into account simultaneously.

Results: We propose a joint deep learning framework for integrating batch effect removal and classification upon various omics data. To this end, we validate it on two private metabolomics (MALDI MS) datasets and one public transcriptomics (scRNA-seq) dataset. Especially for the former, we have achieved the highest diagnostic accuracy (ACC), with notable ~10% improvement than over state-of-the-art methods. Overall, these results indicate that our approach removes batch effect more effectively than conventional methods and yields more accurate

classification results for smart diagnosis.

Introduction

Computational analysis of high-throughput omics data (*i.e.,* genomics, transcriptomics, proteomics, metabolomics and radiomics) has become popular over recent decades[1]. Taking metabolomics as an example, it hunts for quantitative descriptions of complex biological samples (usually urine or blood), and associates clinical observations of diseases with temporal fluctuations of metabolites[2]. By measuring and modeling metabolism alternations in biological samples, metabolomics offers relevance to fresh biological insight into diseases and therapies[2,3]. Similar landscapes also appear in the field of transcriptomics. Single-cell RNA sequencing (scRNA-seq) could identify cell lineages and disentangle cellular heterogeneity in complex tissues by characterizing high-throughput gene expression profiles for cell types and states[4-6], thereby further revealing unexplored biological diversity with valuable pathological information. Many technologies used in biology — including high throughput ones such as microarrays, mass spectrometers and next generation sequencing — depend on a complicated set of reagents and hardware, along with highly trained personnel, to produce accurate measurements[7]. However, batch effects may occur when different technicians are responsible for different subsets of the experiments or diverse reagents, chips or instruments are used. Exactly, batch effects are sub-groups of measurements with qualitatively different behavior across the

conditions unrelated to a study's biological or scientific variables[7]. In matrix assisted laser desorption/ionization mass spectrometry (MALDI MS)[8], for instance, batch effects (if not adequately dealt with) could subsequently lead to serious concerns about the validity of the biological conclusions[7,9] if the serum samples for a patient were repeatedly processed in different plates. Therefore, it is necessary to identify and remove the batch effects before proceeding to the downstream analysis.

It is still a challenge to machine learning[10] to perform computational analysis, considering many measurements (typically corresponding to the feature dimensionality) and usually limited number of samples (or sample size) in high-throughput omics data. The recent leap of deep learning, has outperformed many conventional machine learning techniques at revealing prognostic subtypes in high-risk neuroblastoma[11], analyzing lung adenocarcinoma prognostication[12], and revealing hidden high-resolution cellular subtypes[4]. Research on batch effects is no exception, and some methods based on deep learning have emerged recently. The first application of deep learning in batch effect removal was the ResNet method[13]. While deep network has powerful capability of approximating highly nonlinear mapping, the solution of ResNet is unsupervised (i.e., without knowing the class labels of samples). BERMUDA[4] requires a clustering similarity matrix before training the deep transfer learning network, thus the choice of clustering method and the quality of similarity matrix owns a significant impact on the final results. With the evolution of deep network architecture, there are more and more

breakthroughs in GANs[14] in the past three years. A typical method applied to this field is the NormAE[15] developed by Rong et al. in 2020. Its basic idea lies in constructing an adversarial training procedure between a nonlinear AE to remove batch effects and a discriminator to distinguish the source of domain based on the latent space. However, our original intention is to classify real biological categories, which is essential for diagnostics, prognostics, and identification of metabolic biomarkers. Afterwards, the DESC algorithm invented by Li et al.[5] also relies too much on the auxiliary role of clustering assignment probability. Unfortunately, the classification efficacy would not be improved only by decreasing the mismatching across different batches.

In fact, the rattling bottleneck of batch effect has drawn extensive prior research in the early years, which could split into a variety of different perspectives. First of all, according to the design principle, there are two conventional ways to suppress batch effect, namely location-scale (LS) and matrix-factorization (MF) methods[16]. For instance, ComBat[17] is a popular LS-based approach, which employs a Bayesian framework to model the data by parameterizing location and scale for each batch and feature independently[18]. However, the assumption of normal data distribution for each batch in the LS methods may be over-simplified to treat complicated batch effects as additive and multiplicative components. As an alternative to the LS category, surrogate variable analysis (SVA) provides an MF way to remove batch effect[18]. The MF approaches assumes that the data variation induced by batch effect is independent of the biological interpretation of interest[16,18]. However,

such assumptions may not be valid in practice.

Apart from the above, traditional algorithms could also be divided by application scenarios. For instance, in non-targeted metabolomics diagnosis, there is a need to construct a discriminative model using existing source batches and apply them to predict the labels of future target batches. Ratio_G[19] and fSVA[20] adjust data for enhancement of prediction performance in a predictive model, while most of the conventional studies such as ComBat[17] establish statistical differences at the population level, which are ignorant of the subsequent classification task when modeling batch effect.

Most existing algorithms are based on pairwise analysis in which samples from two batches are considered at a time. Recently, a variety of new methods have emerged in the field of single-cell RNA sequencing (scRNA-seq), including mutual nearest neighbors (MNN)[21], canonical correlation analysis (CCA)[22], and Seurat 3.0[23]. The common point of these methods lies that they all exploit the idea of nearest neighbors to identify the similar clusters between single cells across two batches, thereby integrating them into a shared space. Meanwhile, we also have developed an algorithm recently that specifically designed for improving classification accuracy while removing batch effects. However, it is a pairwise mapping approach in nature. For data with more than two batches, as typically encountered in real-world settings, these approaches could only calibrate pair by pair. Consequently, it is desirable to develop an algorithm that could simultaneously accommodate samples from all batches at once.

To make the subsequent learning-based classification more convincing, we propose a novel deep learning framework to integrate multi-batch calibration and sample classification. We apply the proposed method to two metabolomics diagnosis applications and one transcriptomics classification scenario, respectively, and demonstrate its superior performance in both batch effect removal and classification capability. The contributions of our paper are summarized as follows:

- We borrow the idea of traditional GAN that the discriminator and reconstructor(s) are adversarial to the calibrator in different epoch steps, namely “walk in two steps”, and we also make a breakthrough that the discriminator directly distinguishes accurate biological labels rather than domain information.
- Compared with the framework we designed before, it breaks through the limitation of two batches and could calibrate multiple domains synchronously. Not only does it achieve higher classification accuracy than pairwise ones, but it also more convenient and time-saving.
- Our approach is not only suitable for the binary classification in biological diagnosis of non-targeted metabolomics, but also extends to the multi-classification of cell subtypes under scRNA-seq of transcriptomics.

Results

1. Overview of the method

In this section, we describe our approach that supports batch effect removal for

multiple batches simultaneously. Here we portray the framework, statistical validation, and theoretical deduction of an automated batch adjustment tool designed to minimize batch effects and well suited for better generalization for relevant classification application.

Many works on batch effect removal are applicable to single-source-single-target scenario only. That is, given a source batch where subjects are diagnosed already, one may train a model and then apply it to the target batch. If the batch effect is properly calibrated, the subjects in the target batch can be correctly classified for diagnosis. However, in many real-world settings, one may seek to apply the trained model to multiple (future) target batches, or boost the classification performance by using multiple source batches for training.

Our framework is naturally suitable for such a multi-source-multi-target circumstance. Suppose there are N batches before calibration, $\{\mathbf{X}_1, \cdots, \mathbf{X}_K, \mathbf{X}_{K+1}, \cdots, \mathbf{X}_N\}$, in which $\{\mathbf{X}_1, \cdots, \mathbf{X}_K\}$ represent K source batches of known labels while $\{\mathbf{X}_{K+1}, \cdots, \mathbf{X}_N\}$ are $(N-K)$ unlabeled target batches. The application scenario of our method is to train a model from known labels of source batches, such that the samples in target batches can infer their labels by classification.

For this objective, our first goal is to find a calibrator C to accept all batches of raw data and generate $\{\mathbf{Z}_1, \cdots, \mathbf{Z}_K, \mathbf{Z}_{K+1}, \cdots, \mathbf{Z}_N\}$, where $\mathbf{Z}_i = C(\mathbf{X}_i)$. The calibrated data $\{\mathbf{Z}_1, \cdots, \mathbf{Z}_K, \mathbf{Z}_{K+1}, \cdots, \mathbf{Z}_N\}$ then pass through the discriminator D, which yields the class label per sample. Notice that D is trained only on source batches with its

corresponding class labels $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K$ as supervision, and then we leverage the trained D to classify the corrected samples in target batches and infer predicted labels $\boldsymbol{y}_{K+1}', \ldots, \boldsymbol{y}_N'$, where $\mathbf{y}_i{}' = D(\mathbf{Z}_i)$. Meanwhile, to make sure that the encodes of latent space powerful and well-functioning representation, we enforce all sample data of source batches $\mathbf{Z}_1, \cdots, \mathbf{Z}_K$ to be fully reconstructed from the calibrated latent space (the reason not reconstruct target batches will be confirmed in the experimental part), e.g., by passing them through K individual reconstructors $R_1, \cdots, R_K$ to obtain the reconstruction results $\mathbf{X}_1', \cdots, \mathbf{X}_K'$, namely $\mathbf{X}_i{}' = R_i(\mathbf{Z}_i)$. Figure 1 illustrates the overall framework of our proposed method.

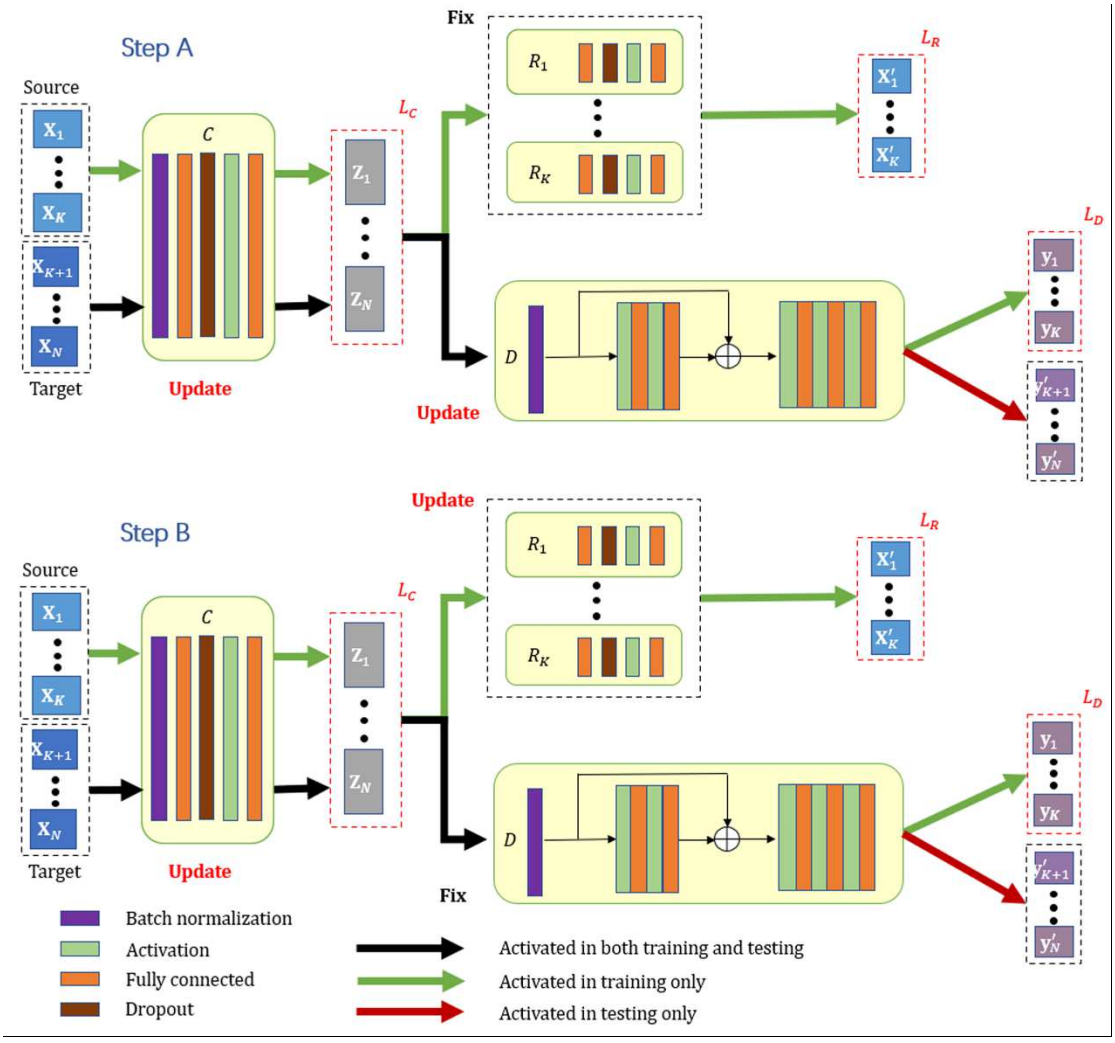


Figure 1. The architecture of our proposed framework and its adversarial training steps. The source batches $\mathbf{X}_1, \cdots, \mathbf{X}_K$ and the target batches $\mathbf{X}_{K+1}, \cdots, \mathbf{X}_N$ are processed through the same calibrator C, to ensure all batches are tightly distributed in the latent space. The source batches supervise the training of discriminator D in step A, which hereafter predicts the class labels for target batches in testing phase. Reconstructors $(R_1, \cdots, R_K)$ are used to recover the input source batches from latent encoding in step B, which guarantees the representative latent features.

In order to make the entire model to converge effectively, two adversarial training steps are involved. In step A, we only renew the model weights of calibrator C and discriminator D without updating the model weights in reconstructor(s) R. Next, we only train C and R with a fixed D in step B. The iterative epochs of the two steps are adjusted according to their respective convergence conditions. With the help of this strategy, the resulted model allows simultaneous alignment multiple batches, from sources to targets.

2. Evaluation metrics

This study proposes a joint deep learning framework to perform multi-batch calibration in multi-category applications. In order to prove in turn that the framework is suitable for multi-source single-target, multi-source multi-target and other omics multi-classification problems, we report experimental results using three datasets from two high-throughput technologies, *i.e.*, two private MALDI MS datasets and one public scRNA-seq dataset. We verify our framework by means of comparing it to several most representative algorithms in the

literature. Detailed assessment will be reported below.

Our evaluation principally focuses on removing batch effect and classification performance. Particularly, for batch effect removal, we adopt MMD as a quantitative metric and present their results in the form of boxplot. We also utilize t-SNE[24,25] to visualize the distribution of the high-dimensional data before and after calibration by each method. Furthermore, we apply four metrics to assess the classification performance, consisting of Accuracy (ACC), F-score, Area Under Curve (AUC), and Matthews correlation coefficient (MCC)[19,26]. The ACC, F-score, MCC are defined below:

$$ACC = \frac{TP + TN}{TP + FN + FP + TN}, \tag{5}$$

$$F\text{-}score = \frac{2 \times TP}{2 \times TP + FP + FN}, \tag{6}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{7}$$

where TP, FP denote the true/false positives and TN, FN denote the true/false negatives.

3. multi-source-single-target situation

Dataset. In order to prove that our approach has obvious utility on multi-source-single-target data, a total of 796 individuals were recruited, including 322 controls and 474 acute myocardial infarction (MI) patients from Shanghai Chest Hospital Affiliated to Shanghai Jiao Tong University. All MI patients had specific histopathological diagnoses (99th cardiac troponin I (cTnI) levels and electrocardiogram (ECG)). The 322 individuals in the control group had no clinical

evidence of cardiovascular diseases such as coronary heart disease (CHD) but may suffer from other diseases. There is no significant difference in age and sex between controls and MI patients. All of the investigation protocols in this study were approved by the institutional ethics committees of the Shanghai Chest Hospital and School of Biomedical Engineering, Shanghai Jiao Tong University. According to the Helsinki Declaration, written informed consent to participate in the study were provided from all individuals, and the use of their biological samples was approved for analysis.

Prove the existence of batch effect. In order to prove that these data are indeed disturbed by batch effect, we first explore the number of differential features in each batch, under which there are significant differences in the expression levels of the case and control groups (e.g., patients and healthy people). We quest the differential peaks for each batch's case and control group according to BPSC[27], a differential expression gene probe algorithm based on Beta-Poisson model. Figure S1a shows that the three batches have only 82 common differential features (about 200 difference peaks are extracted in each batch and take the intersection of three batches) under the condition that the p-value is set to 0.05. It could be concluded that this dataset exists serious batch effects because the differential peaks that separate the case and control group in three batches are distinctive.

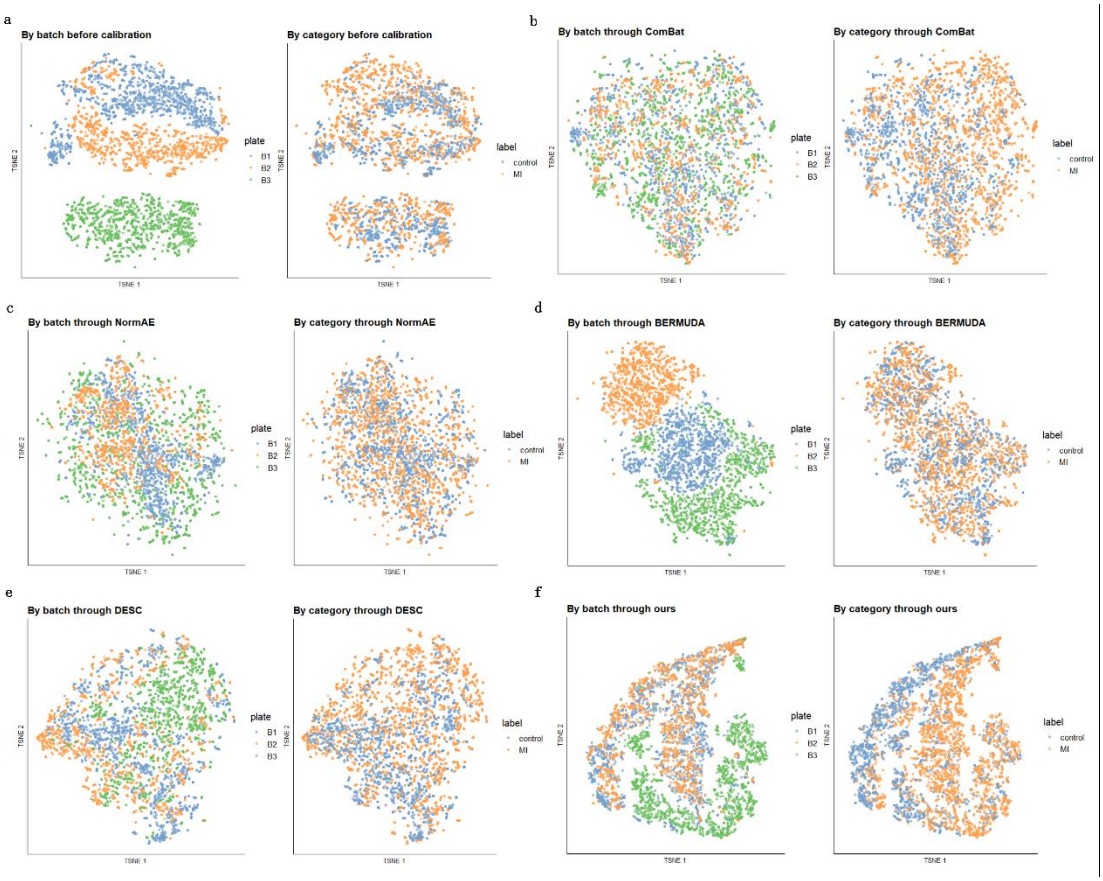


Figure 2. t-SNE Visualization of three batches on the private MI data of MALDI MS. a raw data, **b** ComBat, **c** NormAE, **d** BERMUDA**, e** DESC**, f** our calibration. In the left half of each section, different colors highlight the three batches. In the right half, different colors identify the actual labels of samples (Class 1: MI; Class 0: control).

We further visualize the data distribution before and after batch effect calibration by each algorithm. Figure 2 shows the t-SNE plot of the raw inputs and every calibrated data for 3980 MI samples. Instead of distributing with respect to the batch factor, the same classification label samples should be close to each other in the feature space ideally. However, as in the left part of Figure 2a, one may observe that the three batches (plates, in different colors) are distributed in various patterns, suggesting an apparent batch effect that separates them and may hinder downstream classification (c.f. the right half, colored in accordance to class labels). On the contrary, after being calibrated as in the left part of Figure 2b~2f by each

method, the three batches share thoroughly entangled distributions except BERMUDA (Figure 2d), implying most of their in-between mismatch due to batch effect has been removed. Although ours does not seem to be as uniform as ComBat (Figure 2b), NormAE (Figure 2c) and DESC (Figure 2e) in terms of batch mixing, however, as in the right half of Figure 2f, the samples are naturally more compact in accordance to their labels than other approaches. This is partially due to the discriminator in the two-step strategy, which helps span the feature space to remove batch effect and facilitate the classification of labels.

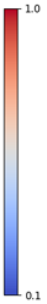
Quantitative comparison. The effectiveness of our framework can further be verified by the classification performance quantitatively. By testing multiple aspects of performance, rather than relying on a single measure, the cross-batch predictions contain $\binom{3}{1}$ classification tasks (choosing 2 out of 3 batches as the training set, and the other batch as the test one) and each batch takes turns once, to test the robustness of our approach. As shown in Table 1a, the quantitative results show that our algorithm can improve in every group, which has increased by as much as 10 percentage points. Especially, the batch effects are strong when batch (1,2) as source and batch 3 as target, the improvement of ACC, F-score, AUC, MCC is as high as 17.2%, 8.2%, 22.1%, 40.2% after applying our method for both endpoints, respectively. Moreover, we also make comparisons with the prior pairwise calibration algorithm developed by ourselves, as shown in the middle column of Table 1a, For each plate, we first use the other two plates to calibrate respectively and then get average. It can be concluded that calibrating multiple

batches simultaneously yields the dual advantages of accuracy and calculation time.

Table 1. Classification results on the MI data of MALDI MS

a

	Before Calibration			Pairwise Calibration			Joint Calibration		
Target	1	2	3	1	2	3	1	2	3
ACC	0.721	0.686	0.599	0.826	0.777	0.754	0.876	0.807	0.771
F-score	0.708	0.693	0.743	0.856	0.828	0.806	0.884	0.853	0.825
AUC	0.745	0.725	0.519	0.814	0.749	0.729	0.886	0.776	0.740
MCC	0.498	0.446	0.128	0.642	0.515	0.487	0.763	0.579	0.530



b

Source	Target	Baseline	ComBat	NormAE	BERMUDA	DESC	Recon_T	Ours	CrossValid
1、2	3	0.664	**0.779**	0.756	0.686	0.775	0.746	0.774	0.937
1、3	2	0.726	0.774	0.708	0.716	0.804	0.751	**0.807**	0.919
2、3	1	0.723	0.795	0.724	0.738	0.756	**0.879**	0.867	0.948
Average		0.704	0.783	0.729	0.713	0.778	0.792	**0.816**	0.935

a The result of four indicators evaluating the cross-batch prediction. The middle column is the result of averaging pairwise calibration, and the right column is the multi-batch calibration developed in this text. **b** Comparison of classification accuracy with multiple source batches for training and only one target batch for testing. Note that "Recon_T" denotes an ablation experiment that reconstruct all target batches.

Afterwards, we select several latest and most representative tools including ComBat[17], NormAE[15], BERMUDA[4], and DESC[5] for further comparison. Accuracy of cross-batch prediction in the sample level is utilized to assess the effectiveness of each method. The comparing results are reported in Table 1b. Our performance is superior over all other approaches, accompanied by an improvement of 3.3~10.3% than others on average. Compared with NormAE, for instance, a novel algorithm tailored for specific metabolomics data types, our method has also

achieved 8.7% improvement (72.9% vs. 81.6% overall) taking advantage of label supervision from the source data. From the results in the third-to-last column, we conclude that the accuracy of reconstructing all batches (including source and target batches) is slightly lower than that for only source batches.

The classification performance could even be comparable to the situation when batch effect is theoretically ruled out. Notably, we compute the in-batch classification accuracy by conducting 10-fold cross-validation within every batch. These results are regarded as a reference to the classification performance without being interfered by batch effect, which are listed in the last column of Table 1b that represent corresponding results within the target batch. We observe that our method produces the results that get much closer to the ceilings where batch effect is completely ruled out.

4. multi-source multi-target situation

Dataset. For further certificating that our algorithm is equally applicable to multi-source and multi-target data, 1203 individuals were recruited in the same way as previous, consisting of 562 healthy controls and 641 coronary heart disease (CHD) patients from the Shanghai Chest Hospital Affiliated to Shanghai Jiao Tong University. For controls, serum samples from 562 healthy volunteers who had no clinical evidence of cardiovascular diseases and other major disease were collected as controls. The same preprocessing pipeline as the MI's was utilized in this experiment. For four batches of this dataset totally, we collected 1559, 1478,

1523, 1449 samples for 312, 296, 305, 290 subjects, respectively.

Prove the existence of batch effect. In this experiment, we prove the existence of batch effect by computing the MMD values between source and target batches before and after calibration by each algorithm. We firstly compute from a subset of 500 samples randomly drawn from all samples available in one batch and then take over 30 permutation runs shown by the form of boxplot. Since it could only be calculated in pairwise, we randomly select batch 3 and batch 4 as an example and show in Figure S1b. Apparently, our calibration decreases the MMD value between the two batches (i.e., 0.152±0.007 after being processed by our method, which is lower than raw data and the results calibrated by other methods). The above results prove that our algorithm can suppress batch effect effectively.

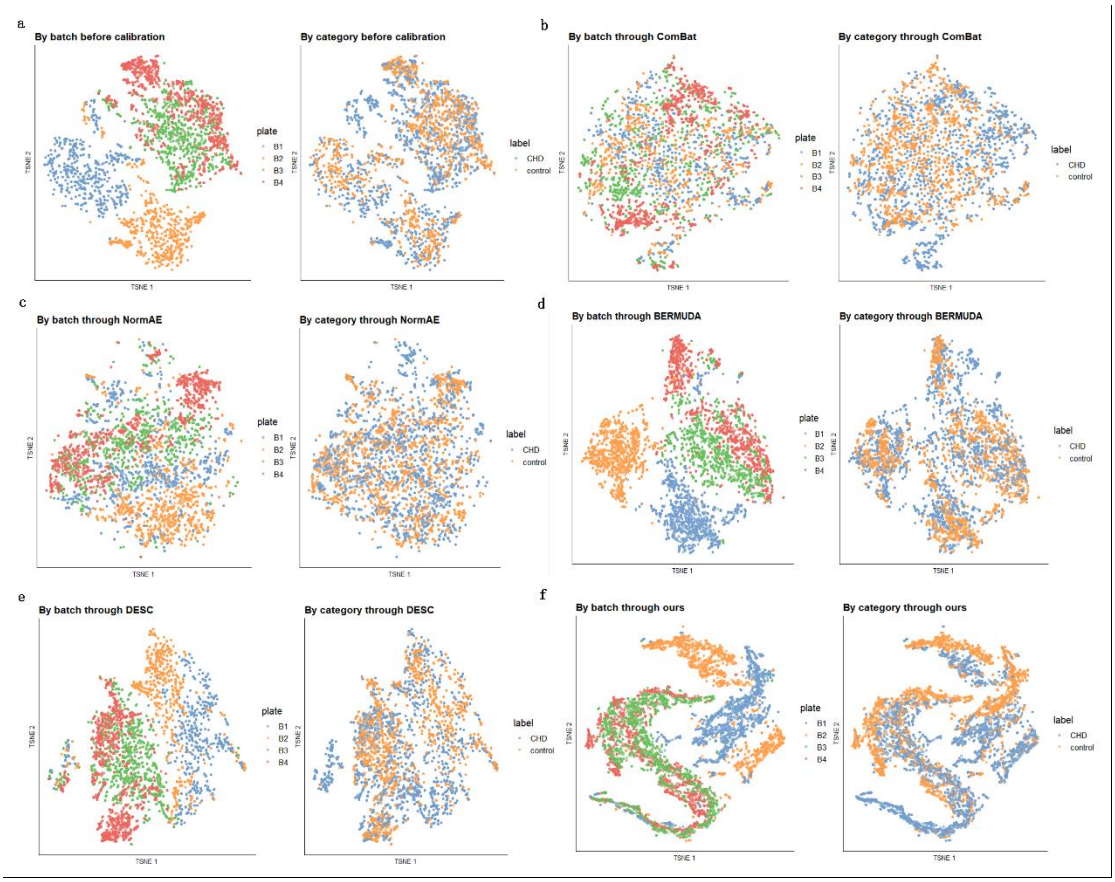


Figure 3. t-SNE Visualization of four batches on the private CHD data of MALDI MS. a raw data, **b** ComBat, **c** NormAE, **d** BERMUDA**, e** DESC**, f** our calibration. The left part of each method is colored by batch indices. In the right half, the samples are colored by disease labels.

We also visualize the distribution by projecting the raw and calibrated data by each method to the same space by t-SNE, as shown in Figure 3. The four batches are mixed together (in the left part of Figure 3f) compared with the case before calibration (the left part in Figure 3a). In this regard, ours is better than BERMUDA (Figure 3d) and comparable to DESC (Figure 3e), but inferior to ComBat (Figure 3b) and NormAE (Figure 3c). On the other hand, if colored by the class labels (CHD vs. control), one may observe that the samples are almost inseparable before calibration (in the right half of Figure 3a), similar scenarios also occur after calibration of other methods (Figure 3b~3e), yet much more separable through our calibration (the right half in Figure 3f). This is exactly what we expect. Such results present different tunes but are rendered with equal skill as the experiments on previous MI data.

Quantitative comparison. We then evaluate the classification performance quantitatively. Given two source batches for training and another two target batches for test, we take the average of different training sets under three cases when each batch as the test set to facilitate intuitive comparison, as shown in Table 2a. Specifically, we also display the comparisons of pairwise calibration in the middle column, which prove that our framework of joint multi-batch calibration is superior to previous pairwise ones in most cases. The original classification results before averaging can be found in Table S1. For example,

when batch (1,2) are used as the training set and batch (3,4) for test, the metrics of ACC, F-score, AUC, and MCC have increased by (13.0%, 19.4%), (3.7%, 6.1%), (16.0%, 19.3%) and (25.6%, 29.9%), respectively.

Table 2. Classification results on the CHD data of MALDI MS

a

	Before Calibration				Pairwise Calibration				Joint Calibration			
Target	1	2	3	4	1	2	3	4	1	2	3	4
ACC	0.732	0.756	0.715	0.639	0.783	0.721	0.715	0.731	0.825	0.815	0.775	0.746
F-score	0.679	0.809	0.768	0.720	0.779	0.755	0.722	0.756	0.820	0.838	0.799	0.752
AUC	0.744	0.735	0.702	0.640	0.787	0.710	0.716	0.732	0.829	0.808	0.770	0.746
MCC	0.527	0.526	0.428	0.334	0.581	0.467	0.439	0.474	0.664	0.631	0.552	0.495

b

Source	Target	Baseline		ComBat		NormAE		BERMUDA		DESC		Recon_T		Ours	
1、2	3、4	0.609	0.559	0.678	0.645	0.521	0.511	0.638	0.665	0.716	0.696	**0.753**	**0.734**	0.751	0.726
1、3	2、4	0.775	0.679	**0.782**	**0.759**	0.662	0.661	0.695	0.691	0.730	0.675	0.683	0.714	0.748	0.718
1、4	2、3	0.715	0.794	0.781	0.787	0.667	0.763	0.743	0.658	0.695	0.792	0.802	0.762	**0.828**	**0.804**
2、3	1、4	0.709	0.659	**0.818**	**0.793**	0.739	0.665	0.702	0.700	0.696	0.744	0.719	0.728	0.769	0.758
2、4	1、3	0.778	0.742	0.771	0.789	0.654	0.687	0.761	0.651	0.764	0.801	0.822	0.799	**0.865**	**0.812**
3、4	1、2	0.649	0.763	0.588	0.578	0.577	0.558	0.599	0.571	0.663	0.752	0.776	0.778	**0.822**	**0.831**
Average		0.703		0.731		0.639		0.673		0.727		0.756		**0.786**	

a The result of four indicators evaluating the cross-batch prediction about before calibration, averaging pairwise calibration, and after multi-batch calibration. **b** Comparison of classification accuracy with multiple source batches for training and multiple target batches for testing. The implication of "Recon_T" is the same as Table 1b.

Next, we still evaluate the methods involved in the previous section. Since the source and target batch are paired, there exists $\binom{4}{2}$ cross-batch prediction tasks. A complete list of the approaches analyzed in this section is provided in Table 2b. The performance of our method is optimal on all experimental groups. Particularly, the last combination, namely batch (3,4) as source and batch (1,2) as target, yields the most improvement for our method (15.9%, 7.9%) compared to the second-ranking approach (DESC). Furthermore, from an ablation experiment in the

penultimate column, the results of reconstructing all batches are inferior to that only reconstructing source batches. These results in overall indicate that our approach not only be suitable for multi-source single-target situations, but also performs well on multi-source multi-target circumstances.

5. scRNA-seq:

Dataset. In order to prove that our method could also achieve better performance in the multi-classification application of other omics besides metabolomics, we focused on the human pancreatic data through several scRNA-seq protocols. We have combined three publicly available datasets generated using CelSeq from Gene Expression Omnibus (GSE81076)[28], Fluidigm C1 (GSE86469)[29] and SMART-Seq2 from the European Bioinformatics Institute (E-MTAB-5061)[30] and each platform represents one batch. The pancreas is a highly heterogeneous tissue with several well-defined cell types. 13 cell type labels have been taken either from the provided metadata or been derived according to the methodology described in the original publication, annotated as “acinar”, “activated stellate”, “alpha”, “beta”, “delta”, “ductal”, “endothelial”, “epsilon”, “gamma”, “macrophage”, “mast”, “quiescent stellate”, “schwann” (further details of data preprocessing as follows).

Prove the existence of batch effect. We also visually compare this data set and the t-SNE results are shown in Figure 4. Notice that in the real-world data, the multiple training and test instances often do not come from the same underlying distribution, but we have projected them onto the same subspace. In the left part

of Figure 4a, the underlying cause of such clustering is the sequencing platform and this is clear because we plot the points marking each point according to protocol numbers. The left part of Figure 4b~4f presents results after calibration by each method, where we can observe that the differences are much smaller after our calibration than ComBat (Figure 4b), NormAE (Figure 4c) and comparable to BERMUDA (Figure 4d), DESC (Figure 4e). In the right half of Figure 4f, the points cluster belonging to the same cell-types appears more compact than before calibration and by most algorithms (the right half in Figure 4a~d), and equal to DESC on this dataset. It implies that the calibration process makes the actual category marker structure significantly more similar.

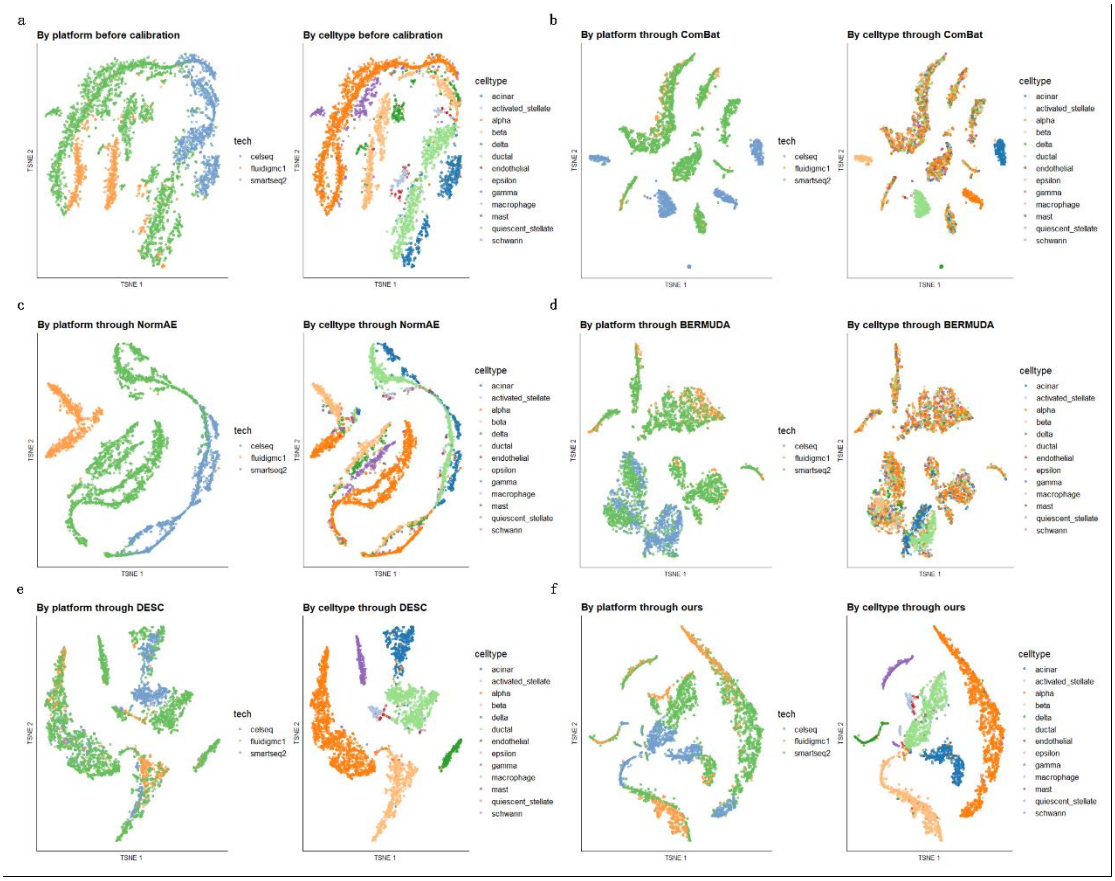


Figure 4. t-SNE visualization of the public pancreas data of scRNA-seq. a raw data, **b** ComBat, **c** NormAE, **d** BERMUDA**, e** DESC**, f** our calibration. In the left part of each section,

different colors highlight the batches (sequencing platforms). In the right half, different colors identify 13 actual labels of cell-types.

Quantitative comparison. In this section, the accuracy of cross-batch prediction is still used to evaluate the effectiveness of those algorithms mentioned above. There are $\binom{3}{1}$ classification tasks and the comparing results are reported in Table 3. Due to a multi-classification task (13 categories), it is much more difficult to improve accuracy than pair-wise classification. Nevertheless, our approach still wins out of the others. Take the second group (batch (1,3) as source and batch 2 as target) as an example, ours outperforms the second-ranking algorithm (NormAE) 24.3 percentage points. In addition, we also make comparisons with the previous pairwise calibration algorithm developed by ourselves, as shown in the "Pairwise" column of Table 3. For each platform, we first use the other two platforms to calibrate separately and then average. These results overall indicate that our approach not only be more suitable for binary classification diagnosis of metabolomics than other methods, but also suitable for multi-classification scenarios of other omics.

Table 3. Classification results on the pancreas data of scRNA-seq

Source	Target	Baseline	ComBat	NormAE	BERMUDA	DESC	Pairwise	Recon_T	Ours
1、2	3	0.162	0.147	0.432	0.187	0.135	0.233	0.321	0.428
1、3	2	0.226	0.544	0.215	0.128	0.208	0.183	0.370	0.458
2、3	1	0.001	0.161	0.221	0.118	0.163	0.165	0.050	0.227
Average		0.130	0.284	0.289	0.144	0.169	0.194	0.247	0.371

Comparison of classification accuracy with multiple categories on the scRNA-seq of transcriptomics. Note that "Pairwise" represents the two-batch calibration method designed by ourselves.

Discussion

In recent decades, high throughput omics analysis technology has become much more mature. However, batch effects are ubiquitous in high-throughput experiments (e.g., RNA sequencing, metabolomics), the source of which is far-ranging, including different platforms, different periods of the same platform, different reagents and times of the same sample, etc. To this end, we introduce an end-to-end deep learning framework and demonstrate that our proposed method can effectively remove batch effect by the experiments on two metabolomics and one transcriptomics datasets. Our framework outperforms all compared methods in terms of classification accuracy through the experiments. This is partially attributed to multiple modules that are interacted with each other in the architecture. In addition, the "two-step" adversarial strategy we adopted during training also dramatically facilitates the compromise between the classification and reconstruction modules, something other algorithms fails to account for. The mutual restraint of these modules is substantially improving the overall performance of our network.

A considerable number of computational methods in genomics and transcriptomics have been developed to remove batch effects. However, they might be less effective in improving the accuracy of classification based on different omics, because it is not easy to generalize their findings to the sample

level for the sake of individualized biological diagnosis and treatment[31]. Although NormAE is based on metabolomics, it is not superior to others in MALDI MS diagnosis. A possible reason lies that it doesn't take advantage of the supervisory information provided by the source labels. DESC takes subsequent classification tasks into account, so it can yield better performance than NormAE or BERMUDA. Although DESC is an adversarial model and the application scenario also involves classification, and BERMUDA utilizes the deep transfer learning network, both depend too much on the clustering similarity matrix. Overall, those methods are intrinsically driven by data size/scale that is susceptible to bias introduced by batch effects and cannot effectively address these deviations especially concerning the need for classification.

There are some deficiencies in this work that cannot be underestimated. In the cases where exist multiple cell types in transcriptome, the improvement of some batches is not apparent. Generally, assumptions made by learning algorithms that identical cell types in different batches are often violated, resulting in degradation of the algorithms' performance during inference of test data. We might as well calculate the distribution distance of test set with those labels-known source data and choose the closest ones as training set under this circumstance from now on. Furthermore, conclusions reached in this study are based on the application of batch effect removal in the context of cross-batch prediction. Beyond the fact that this approach is no longer unsupervised and requires domain knowledges, the amount of labeled data that might be needed to achieve reasonable performance

could be large. These are still outstanding questions and need further investigation.

Conclusion

The rise of omics techniques has resulted in an explosion of high throughput data in modern biomedical research. However, these analytical barriers are further compounded, since the bias introduced by the non-biological nature of the batch effect can be strong enough to mask, or confound actual biological differences. Therefore, it is necessary to develop a novel tool, which will be time efficient, incorporate flexibility for data types, investigator-driven batch adjustment approach choices, and the ability to evaluate such adjustment approaches, with great application potential for the analysis of large samples in clinical studies.

We have introduced a novel end-to-end learning framework for simultaneous multi-batch calibration and classification, and we conduct adversarial training by "walk-in two steps" strategy. Upon the private MALDI MS and public scRNA-seq datasets, we confirm that our framework can effectively suppress batch effect and accomplish classification, outperforming the second-ranking algorithm by a substantial margin on many groups. We have witnessed the applications to transcriptomics and metabolomics datasets here and released publicly available codes on the GitHub. Furthermore, in the same way that general deep learning techniques, operating on raw data outperform traditional algorithms tailored for specific data types, involving domain knowledge, or massive pre-processing, we

demonstrate that our proposed algorithm and experimental results really promising and may open new horizon for removing batch effects in biological datasets.

Methods

1. Loss Function

Calibration loss MMD. Our calibrator C is responsible for lessoning the discrepancy between source and target batches by matching them into a common space. It comprises a batch normalization (BN), a fully connected (FC), a dropout layer, a Tanhshrink activation and again an FC layers. The number of nodes in each hidden layer remains the same as the feature dimensionality of inputs. In addition, it is critical to define a measurement of divergence among source and target batches' distributions. Therefore, we train the calibrator C by minimizing the maximum mean discrepancy (MMD) in the latent space:

$$\mathcal{L}_C = \binom{N}{2}^{-1} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left\| C(\mathbf{X}_i) - C(\mathbf{X}_j) \right\|_1, \quad (1)$$

where $\mathbf{X}_i$ and $\mathbf{X}_j$ indicate the i-th source batch and the j-th target batch respectively, and $\|\cdot\|_1$ is the L1-norm operator. MMD value approaches zero when the underlying distributions of the observed samples are highly similar.

Reconstruction loss MSE. In order to prevent losing the intrinsic biological states encoded in the latent code $\mathbf{Z}$, we require $R(\mathbf{Z})$ to transform the encodes $\mathbf{Z}$ back to the reconstructed $\mathbf{X'}$ that reflects the characteristics of the original data. All of reconstructors have the same structure that consists of first and last two FC layers,

a dropout layer and a Tanhshrink activation. The reconstruction loss was calculated as the residual in L2-norm between the input prior to calibration and the reconstructed output of the encoder-decoder backbone for self-learning:

$$\mathcal{L}_R = \frac{1}{K}\sum_{i=1}^{K}\left\|\mathbf{X}_i - R_i\left(C(\mathbf{X}_i)\right)\right\|_2^2. \quad (2)$$

Note that K denotes the number of source batches and each reconstructor corresponds to a certain batch.

Classification loss CE. In addition, we introduce D to be a task-specific class label classifier. It should be noted that we distinguish the category labels for the source batches in the training stage. The class label information of the target batches is only used when evaluating the prediction performance.

The internal structure of discriminator D has similar components but different arrangement with C or R. Specifically, it involves three partitions: the first one only contains a BN layer; the second partition is a residual structure, including two alternating Leaky ReLU activation layers and FC whose number of nodes switching between input scale and 100; the last one contains three alternating Leaky ReLU activation layers and FC, in which the node reduced from input scale to 50, 16, and finally to the number of categories. When the application scenario is only binary classification considered, the last layer utilizes the Sigmoid activation function. We calculate the loss as cross-entropy (CE) between its output and sample biological labels:

$$\mathcal{L}_D = -\frac{1}{K}\sum_{i=1}^{K}\sum_{j}\sum_{c=1}^{M}\boldsymbol{y}_{jc}\log D\left(C\left(\mathbf{x}_{jc}\right)\right). \quad (3)$$

where K and M represent the number of batches or categories respectively, and

$\boldsymbol{y}_{jc}$ can be assigned 1 when the true category of sample j is equal to c, otherwise 0.

2. Training Process

In order to control the iterative epochs in training the discriminator and reconstructor(s) more conveniently, we adopt a "two-step" strategy in the training process. We set two hyperparameters λ_1, λ_2 to control the step A and step B respectively, as shown in formula (4). Note that the parameters of calibrator C are always kept updating. The benefits of doing so lies that when the reconstruction converges slowly, we increase the number of rounds of step A, and on the contrary, increase the number of iterations of step B. These two steps are repeated back and forth until reaching convergence of the model. The total loss of our framework is then calculated by considering the calibrator, reconstructor, and discriminator as a whole, i.e., to minimize

$$\mathcal{L} = \lambda_1 \cdot (\alpha_1 \cdot \mathcal{L}_C + \beta \cdot \mathcal{L}_D) + \lambda_2 \cdot (\alpha_2 \cdot \mathcal{L}_C + \gamma \cdot \mathcal{L}_R), \quad (4)$$

where λ_1, λ_2 denote the number of reincarnations aiming at step A and step B, and α_1, β, α_2, γ are scalar weights for each component network in two steps, respectively.

We implement the proposed scheme with PyTorch (version 1.8.1+cu102) and Sklearn (version 0.21.3) framework. The downstream analysis has been carried out using Python (version 3.6.8), and R (version 4.0.4) for visualization. For details, we use ADAM for training with default settings (i.e., the exponential decay rate of the first/second moment estimation). To achieve fair comparison, the types and

numbers of classification network layers remain the same before and after calibration. The gradient update rule of mini-batch in deep learning is used to train our model where corresponding losses are calculated from a sampled "mini-batch" during each iteration of the corresponding step. All the experiments are run on the same host with 16GB memory and an Nvidia RTX 2080Ti GPU.

3. Information for Other Methods

Corresponding open source code could be found about those algorithms involved in comparative experiments. The ComBat has been implemented by ComBat() function into R software package sva (http://bioconductor.org/packages/3.5/bioc/html/sva.html). The source codes of NormAE algorithm are publicly available at https://github.com/luyiyun/NormAE. Since our data based MALDI MS or scRNA-seq instead of LC MS in NormAE, which not exist so-called injection order, therefore, it is eliminated in training and testing process. In addition, the mass quality control was conducted using standard molecules on the stage of serum plates, so it doesn't appear at the preprocessing matrix. The source codes of BERMUDA are publicly available at https://github.com/txWang/BERMUDA. In principle, we compute the similarity matrix based on the MetaNeighbor algorithm by following recommended protocols in its reports. An open-source implementation of the DESC can be downloaded from https://eleozzr.github.io/desc/. We generally yield a .h5ad file through preparing an AnnData object based on the processing pipeline for following analysis.

4. MI Dataset

The collection process for the MALDI MS data was introduced by Huang, et al.[32]. According to the protocol, ~2 mL of blood was collected by venipuncture and centrifuged at 3000 rpm for 10 min. Then, the serum was transferred to a microtube and stored at -80℃. For each subject, we repeated detection five times to enhance the reproducibility and stability. The samples with a relative standard deviation (RSD) less than 5% were discarded in subsequent quality control. Consequently, each subject would eventually yield 1-5 (mostly 5) samples. While each sample could have a predicted label in test, the diagnosis result should be ensembled to the subject level as the median of all samples in one subject. In total, for the three batches, we collected 1330, 1305, 1345 samples for 266, 261, 269 subjects, respectively. Note that the following results were all reported at the sample level. All data was processed through smoothing filter, baseline correction, peak extraction and peak alignment[32]. As for each sample, the m/z range was set from 100 to 300[33], and about 200 features were extracted through data preprocessing.

In the experiment of classification before calibration, the learning rate, number of epochs, size of mini-batch and learning step are set to 10^{-4}, 100, 128 and 10^{4} respectively. In order to prevent overfitting the network, the L2 weight decay is set to 5×10^{-5} during training. For training our network after calibration, the learning rate is set to 10^{-3} and the number of epochs is set to 100. We set the mini-batch size to 128 and the coefficients of losses are $\alpha_1 = 10$, $\beta = 10$ in step

A for $\lambda_1 = 1$ and $\alpha_2 = 10$, $\gamma = 0.01$ in step B for $\lambda_2 = 1$ by grid-search. The learning step, L2 weight decay are set to 10^4, 5×10^{-5} during training. In order to compare all the results on the same benchmark, the classification network of in-batch 10-fold cross-validation shares the same framework as cross-batch experiments. The learning rate, number of epochs, size of mini-batch, learning step and L2 weight decay are set to 10^{-3}, 10, 128, and 100,5×10^{-5}, respectively. For most of other methods, including ComBat, BERMUDA and DESC, we evaluate their performance based on the default parameters and by following recommended pipeline in their tutorials. As for NormAE, in order to ensure the convergence of the model, except for the (lr_rec, lr_disc_b), epoch and batch_size which are set to (0.0002, 0.0001), (200, 100, 150) and 200, other parameters are defaulted.

5. CHD Dataset

In this set of experiment, the structure of classification network is also consistent with the previous rule, namely keeping the same before and after calibration. The learning rate, number of epochs, size of mini-batch, learning step and L2 weight decay are set to 10^{-4}, 50, 128, 10^4 and 5×10^{-5}, respectively. For parameters after calibration, the learning rate, number of epochs, size of mini-batch and learning step are set to 10^{-4}, 50, 128 and 10^4, respectively. In addition, the hyper-parameters of the coefficients of losses are set to $\alpha_1 = 10$, $\beta = 10$ in step A for $\lambda_1 = 4$ and $\alpha_2 = 10$, $\gamma = 0.01$ in step B for $\lambda_2 = 1$ by grid-search. The L2 weight decay is set to 5×10^{-5} during training, the same with first experiment. We

evaluate most of other methods except NormAE based on the default parameters and tutorials as prior. As for NormAE, the (lr_rec, lr_disc_b), epoch and batch_size are set to (0.0002, 0.0001), (100, 50, 100), 200, and other parameters are defaulted.

6. scRNA-seq Dataset

The data batches were introduced in the form of Seurat[23] R objects featuring standardized annotations, which relies on anchor cells between pairs of datasets. However, misidentification of anchors from different batches might have led to reduced classification accuracy. Therefore, we need segment objects according to different protocols and transform them into Single Cell Experiment (SCE) objects. Metadata and counts were extracted from the SCE R objects and used for performing standard preprocessing. Next, we explored common genes by "scran" R package and only genes that were detected in all three experiments were kept. Through a series of procedures such as filtering low-quality cells, standardizing, and selecting the most informative genes by calculating the degree of variation for each gene, the resulting dataset consists of three batches for a total of 4036 cells with 325 genes each.

For classification before calibration, the learning rate, number of epochs, size of mini-batch, learning step and L2 weight decay are set to 10^{-5}, 50, 128, 10^{4} and 5×10^{-5}, respectively. For framework after calibration, the learning rate, number of epochs, size of mini-batch, learning step and L2 weight decay are set to 5×10^{-6}, 55, 128, 10^{4} and 5×10^{-5}, respectively. Moreover, the hyper-parameters of the coefficients of losses are set to $\alpha_1 = 10$, $\beta = 10$ in step A for

$\lambda_1 = 1$ and $\alpha_2 = 10$, $\gamma = 0.1$ in step B for $\lambda_2 = 1$. The ComBat, BERMUDA and DESC still utilize the default parameters and tutorials. In NormAE, the (lr_rec, lr_disc_b), epoch and batch_size are set to (0.0002, 0.0001), (100, 50, 100), 200, while other parameters are yet defaulted.

Abbreviations

scRNA-seq ,	single-cell RNA sequencing
MS,	mass spectrometry
MALDI,	matrix assisted laser desorption/ionization
SVA,	surrogate variable analysis
MMD,	maximum mean discrepancy
MSE,	mean square error
AE,	Autoencoder
GAN,	Generative Adversarial Network
MCC,	Matthews correlation coefficient
MI,	myocardial infarction
CHD,	coronary heart disease

References

1 Md. Mohaiminul Islam, Y. W. a. P. H. Deep Learning Models for Predicting Phenotypic Traits and Diseases from Omics Data. *Artificial Intelligence - Emerging Trends and Applications*, doi:10.5772/intechopen.75311 (2018).

2 Nicholson, J. K. & Lindon, J. C. Metabonomics. *Nature* **455**, 1054-1056, doi:10.1038/4551054a (2008).

3 Zenobi, R. Single-Cell Metabolomics: Analytical and Biological Perspectives. *Science* **342**,

1243259, doi:10.1126/science.1243259 (2013).

4 Wang, T. *et al.* BERMUDA: a novel deep transfer learning method for single-cell RNA sequencing batch correction reveals hidden high-resolution cellular subtypes. *Genome Biol* **20**, 165, doi:10.1186/s13059-019-1764-6 (2019).

5 Li, X. *et al.* Deep learning enables accurate clustering with batch effect removal in single-cell RNA-seq analysis. *Nature Communications* **11**, 2338, doi:10.1038/s41467-020-15851-3 (2020).

6 Stegle, O., Teichmann, S. A. & Marioni, J. C. Computational and analytical challenges in single-cell transcriptomics. *Nat Rev Genet* **16**, 133-145, doi:10.1038/nrg3833 (2015).

7 Leek, J. T. *et al.* Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet* **11**, 733-739, doi:10.1038/nrg2825 (2010).

8 Cohen, S. L. & Chait, B. T. Influence of Matrix Solution Conditions on the MALDI-MS Analysis of Peptides and Proteins. *Analytical Chemistry* **68**, 31-37, doi:10.1021/ac9507956 (1996).

9 Akey, J. M., Biswas, S., Leek, J. T. & Storey, J. D. On the design and analysis of gene expression studies in human populations. *Nature Genetics* **39**, 807-808, doi:10.1038/ng0707-807 (2007).

10 Zhang, Z. *et al.* Deep learning in omics: a survey and guideline. *Brief Funct Genomics* **18**, 41-57, doi:10.1093/bfgp/ely030 (2019).

11 Zhang, L. *et al.* Deep Learning-Based Multi-Omics Data Integration Reveals Two Prognostic Subtypes in High-Risk Neuroblastoma. *Frontiers in Genetics* **9**, 477, doi:10.3389/fgene.2018.00477 (2018).

12 Lee, T.-Y., Huang, K.-Y., Chuang, C.-H., Lee, C.-Y. & Chang, T.-H. Incorporating deep learning and multi-omics autoencoding for analysis of lung adenocarcinoma prognostication. *Computational Biology and Chemistry* **87**, 107277, doi:https://doi.org/10.1016/j.compbiolchem.2020.107277 (2020).

13 Shaham, U. *et al.* Removal of batch effects using distribution-matching residual networks. *Bioinformatics* **33**, 2539-2546, doi:10.1093/bioinformatics/btx196 (2017).

14 Goodfellow, I. J. *et al.* in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2* 2672–2680 (MIT Press, Montreal, Canada, 2014).

15 Rong, Z. *et al.* NormAE: Deep Adversarial Learning Model to Remove Batch Effects in Liquid Chromatography Mass Spectrometry-Based Metabolomics Data. *Anal Chem* **92**, 5082-5090, doi:10.1021/acs.analchem.9b05460 (2020).

16 Lazar, C. *et al.* Batch effect removal methods for microarray gene expression data integration: a survey. *Brief Bioinform* **14**, 469-490, doi:10.1093/bib/bbs037 (2013).

17 Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **8**, 118-127, doi:10.1093/biostatistics/kxj037 (2007).

18 Leek, J. T. & Storey, J. D. Capturing Heterogeneity in Gene Expression Studies by Surrogate Variable Analysis. *PLOS Genetics* **3**, e161, doi:10.1371/journal.pgen.0030161 (2007).

19 Luo, J. *et al.* A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *Pharmacogenomics J* **10**, 278-291, doi:10.1038/tpj.2010.57 (2010).

20 Parker, H. S., Corrada Bravo, H. & Leek, J. T. Removing batch effects for prediction problems with frozen surrogate variable analysis. *PeerJ* **2**, e561, doi:10.7717/peerj.561 (2014).

21 Haghverdi, L., Lun, A. T. L., Morgan, M. D. & Marioni, J. C. Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat Biotechnol* **36**, 421-427, doi:10.1038/nbt.4091 (2018).

22 Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nat Biotechnol* **36**, 411-420, doi:10.1038/nbt.4096 (2018).

23 Stuart, T. *et al.* Comprehensive Integration of Single-Cell Data. *Cell* **177**, 1888-1902.e1821, doi:10.1016/j.cell.2019.05.031 (2019).

24 Laurens van der Maaten, G. H. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9**, 2579-2605 (2008).

25 Maaten, L. V. D. Accelerating t-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **15**, 3221–3245 (2014).

26 Shi, L. *et al.* The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nat Biotechnol* **28**, 827-838, doi:10.1038/nbt.1665 (2010).

27 Vu, T. N. *et al.* Beta-Poisson model for single-cell RNA-seq data analyses. *Bioinformatics* **32**, 2128-2135, doi:10.1093/bioinformatics/btw202 (2016).

28 Muraro, Mauro J. *et al.* A Single-Cell Transcriptome Atlas of the Human Pancreas. *Cell Systems* **3**, 385-394.e383, doi:https://doi.org/10.1016/j.cels.2016.09.002 (2016).

29 Lawlor, N. *et al.* Single-cell transcriptomes identify human islet cell signatures and reveal cell-type-specific expression changes in type 2 diabetes. *Genome Res* **27**, 208-222, doi:10.1101/gr.212720.116 (2017).

30 Segerstolpe, Å. *et al.* Single-Cell Transcriptome Profiling of Human Pancreatic Islets in Health and Type 2 Diabetes. *Cell Metabolism* **24**, 593-607, doi:https://doi.org/10.1016/j.cmet.2016.08.020 (2016).

31 Yamada, R., Okada, D., Wang, J., Basak, T. & Koyama, S. Interpretation of omics data analyses. *Journal of Human Genetics*, doi:10.1038/s10038-020-0763-5 (2020).

32 Huang, L. *et al.* Machine learning of serum metabolic patterns encodes early-stage lung adenocarcinoma. *Nat Commun* **11**, 3556, doi:10.1038/s41467-020-17347-6 (2020).

33 Qiu, Y. *et al.* Serum Metabolite Profiling of Human Colorectal Cancer Using GC−TOFMS and UPLC − QTOFMS. *Journal of Proteome Research* **8**, 4844-4850, doi:10.1021/pr9004162 (2009).

Authors' contributions

Conceptualization: J.N., Q.W. Methodology: J.N., Q.W. Acquisition of data and materials: W.X., K.Q. Formal analysis: J.N., Q.W. Writting: J.N. Review and editing,

D.W. Supervision: D.W., Q.W. Funding acquisition: Q.W. All authors approved the final version.

Funding

Not applicable

Availability of data and materials

Complete code and part of data are publicly available at: https://github.com/n778509775/NWCQ.git

Ethics approval and consent to participate

Not applicable

Competing interests

Not applicable

Author details

[1]School of Biomedical Engineering, Shanghai Jiao Tong University, Shanghai, 200030, China

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementaryinformation.pdf