Study of viable strategies for combating batch effects in high-throughput data

Supervisor： Prof. Wilson Wen Bin Goh
Prof. Andrew Chi-Hau Sue
Student name: Longjian Zhou

ABSTRACT

The presence of technical confounders (also referred to as batch effects), impedes high-throughput/big data research by causing both false positives and false negatives. This in turn, creates severe problems for translational and pharmacological applications such as gene signature inference, predictive modeling and intervention (drug/treatment) analysis. The subject of this thesis is studying viable strategies for combating batch effects in high-throughput data, focusing on evaluating the practical limitation of batch effect correction algorithms and providing viable strategies for correcting batch effects to this field. The main contents are list as follows:

In chapter 1, we provided a current overview of how batch effects are caused and commonly used batch effects detection algorithms, batch effects correction algorithms, and the exigent problems. We concluded that the problem of batch class confounding are particularly poignant and theorized how methods can resolve it.

In chapter 2, we examined the practical limits of batch effect-correction algorithms: When should you care about batch effects? Several statistical approaches have been devised for resolving batch effects. Although many performance ranking exercises have been conducted to establish the best batch effect-correction algorithm (BECA), we hold the viewpoint that the notion of best is context-dependent. Moreover, alternative questions beyond the simplistic notion of “best” are also interesting: are BECAs robust against various degrees of confounding and if so, what is the limit? Using two different methods for simulating class (phenotype) and batch effects and taking various representative datasets across both transcriptomic and proteomic platforms, we demonstrate that under situations where sample classes and batch factors are moderately confounded, most BECAs are remarkably robust and only weakly affected by upstream normalization procedures. This observation is consistently supported across the multitude of test datasets. BECAs do have limits: When sample classes and batch factors are strongly confounded, BECA performance declines, with variable performance in precision, recall and also batch correction. We also report that while conventional normalization methods have minimal impact on batch effect correction, they do not affect downstream statistical feature selection, and

in strongly confounded scenarios, may even outperform BECAs. In other words, removing batch effects is no guarantee of optimal functional analysis. Overall, this study suggests that simplistic performance ranking exercises are quite trivial, and all BECAs are compromises in some context or another.

In chapter 3, we proposed a class-specific ComBat strategy and demonstrated it more robust against batch-class confounding issues than existing ComBat strategy. ComBat is commonly used as a batch effect-correction algorithm (BECA). However, when using it to do batch-effect correction, batch information is presented but class (phenotype) information is generally ignored. In situations where batch and class effects are confounded due to experimental design imbalances (batch-class confounding), this may lead to performance issues. We propose an alternative procedure for performing ComBat (class-specific ComBat; CS-ComBat), where batch effects on each sample class are corrected independently before being merged. We examine CS-ComBat in conjunction with two other strategies of performing ComBat and with other BECAs (Ratio-A, Ratio-G, BMC). We demonstrate that, given batch-class confounding and with medium to high class-effect proportion (CEP), good performance based on batch effect correction, class effect preservation, inter-sampling similarity and statistical feature selection can be achieved by CS-ComBat. We also show that, given batch-class confounding, the rebalancing approach Synthetic Minority Oversampling TEchnique (SMOTE) synergizes with BECAs for effective batch effect removal while preserving class effects with medium to high CEP. In summary, ComBat is a powerful approach, but blindly applying it on whole data without considering batch-class confounding issues is ill-advised. CS-ComBat is a candidate approach for dealing with batch-class confounding cases when CEP is high. In addition, ComBat and other BECAs seem to benefit substantially from the use of SMOTE when the batch-class is confounded.

In chapter 4, we investigated class-specific ComBat for correcting batch effects in high-throughput data with a focus on small sample size. Due to time, expense, and sample limitations, many high-throughput data samples are relatively small, and the existing batch effect algorithms are not suitable for small samples size. ComBat is one of the few algorithms that can be used to process small sample size data. In the previous chapter, we examined the limitations of ComBat and proposed a CS-ComBat strategy and evaluated in the data with a larger sample size. But it performance on the

small sample size has not been evaluated. In this chapter, we focused on evaluating the performance of CS-ComBat, ComBat and ComBat (cov) under small sample size scenario. Across both test genomics datasets with real and simulated batch effects under small sample size scenarios, CS-ComBat consistently removes batch effects more thoroughly and provides better recall during statistical feature selection.

In chapter 5, we considered batch effect correction on proteomics data: correcting at peptide level or protein level? High-throughput proteomics facilitates large-scale protein characterization. However, proteomics data are often suffering from batch effects problems. Whereas many studies focus on correcting batch effects at the protein level, the analysis at earlier peptide-level is seldom studied, and it is unclear if batch effects correction done earlier in the peptide level, would the final protein level be more accurate. In this study, we comprehensively evaluated batch effects correction at peptide level and protein level on proteomics data that originated from advanced proteomics technology (SWATH). Principal component analysis (PCA) and partial redundancy analysis (pRDA) are used to detect the batch effects. Jaccard coefficient was used to evaluate inter-sampling similarity. And statistical feature selection followed by precision, recall and F-score was conducted to evaluate the overall performance. Our findings suggested that batch effects correction at peptide level perform as well as batch effects correction at protein level, when to use which is based on analytical needed.

In chapter 6, the work of this thesis is summarized and prospected, and the innovation of this paper is emphasized.

In summary, this thesis comprehensively studied viable strategies for batch effects correction in high-throughput data. These findings will facilitate researchers to pay more attention to the batch effects issue and alleviate the impaction of batch effects to their experiment, and provide potential application in identifying gene signature and drug-target.

KEY WORDS： High-throughput data; Batch effects; Batch effect-correction algorithm; Bioinformatics; Feature selection; Normalization; Statistics

CONTENTS

Chapter 1. Background

1.1 overview

Thanks to new high-throughput technologies, testing thousands or millions of biological features simultaneously become a reality, producing a large amount of high-dimensional molecular information (Omics data)[1-4]. Omics data (e.g., genomics, transcriptomics, and proteomics) combined with statistical feature selection (SFS), focuses on identifying and containing the most relevant variables among all other measurements[5]. It offers potential towards understanding the complex biological mechanisms underpinning disease[6, 7], diagnostics[8], prognostics[9, 10] and therapeutics[11]. However, in analyzing omics data, one key issue is technical bias (batch effects)[5, 12-17](Fig. 1-1).

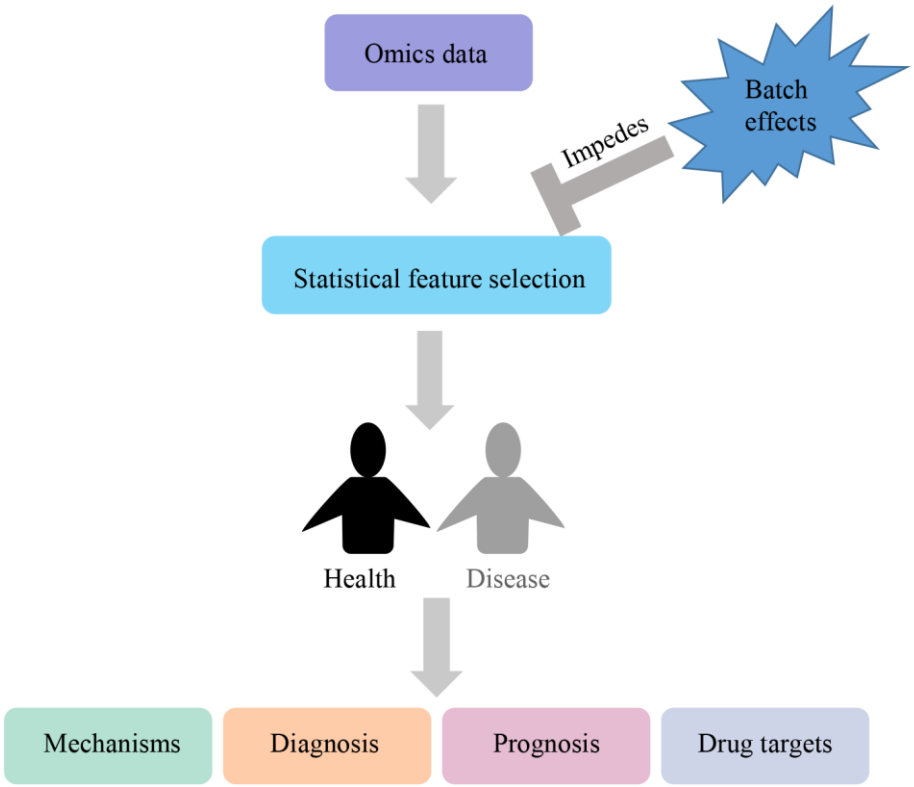


Fig. 1-1. High-throughput data combined with statistical feature selection with various important applications, but the effective use of statistical feature selection was impeded by batch effects.

Batch effects are technical bias that confounds the detection of real biological signals. It may be caused by different processing times, handling personals, reagent lots, platforms, etc[15, 16, 18, 19]. Batch effects almost unavoidable due to the practical complications and sample limitation of the high-throughput technologies: for example, the 1000 Genomes Project[20], need larger samples and cooperation across several countries, and performed at different times, places, and platforms[12, 17]. The downstream analyses of omics data without batch effects correction leads to misleading results[12, 15] or decrease in the statistical power[21, 22]. Therefore, batch effects need to deal with satisfactorily[5, 17, 23].

In the past ten years, the impact of batch effects on high-throughput omics data has received more and more attention. Through the statistics of the papers on batch effects published on the web of science (Fig. 1-2), the number of publications has been rising continuously in the last 10 years in general. This is not surprising because this question provides many exciting directions for computational scientists and biologists. For computational scientists, it is fascinated with developing new and improved methods to better deal with this problem. For biologists, it is very attractive to combine their own data with publicly available databases (e.g. Gene Expression Omnibus GEO[24], ArrayExpress[25], TCGA[26], and many other databases see Table 1-1) to improve their statistical power and gain maximum biological insight.

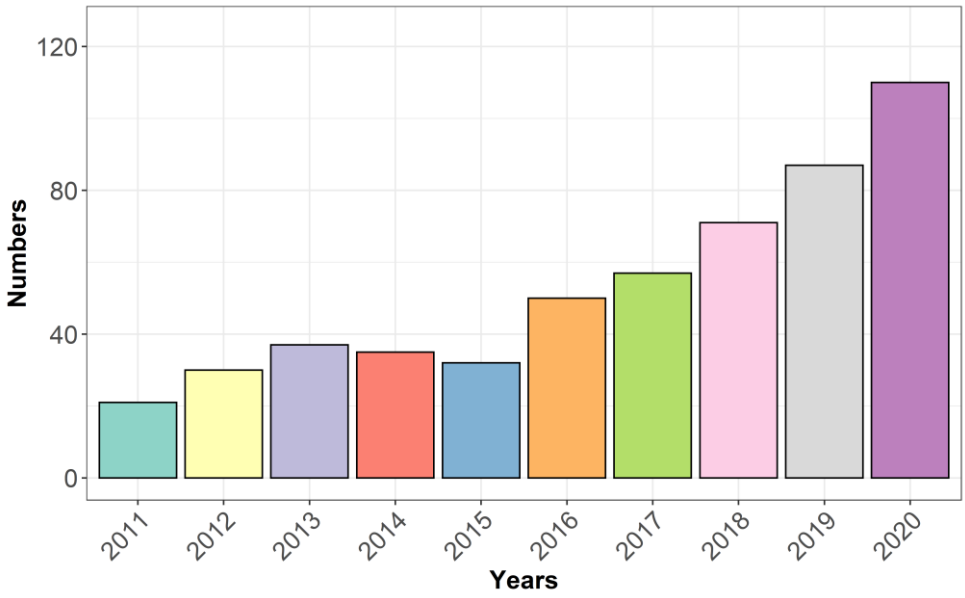


Fig. 1-2. Research progress of batch effects in the last 10 years. The X-axis indicates the years from 2011 to 2020. The Y-axis shows the numbers of publications found in Web of Science on the topic of batch effects, archived until December 31, 2020.

Table 1-1. Commonly used databases for retrieving and or storing high-throughput data.

Database	Features	Website	Reference
Gene expression Omnibus (GEO)	The largest and most comprehensive public gene expression data resource.	https://www.ncbi.nlm.nih.gov/gds/	[24]
ArrayExpress	A public functional high throughput gene expression data repository.	https://www.ebi.ac.uk/arrayexpress/	[25]
The Cancer Genome Atlas (TCGA)	Characterizing over 20,000 primary cancer.	https://www.cancergenome.nih.gov/	[26]
Oncomine	A cancer microarray database.	https://www.oncomine.org/resource/login.html	[27]
1000 Genomes Project	Cataloging normal variation in diverse human populations.	www.1000genomes.org	[20]
Encyclopedia of DNA Elements (ENCODE)	Identifying functional genomic elements in the human genome.	https://www.encodeproject.org/	[28]
Genomics of Drug Sensitivity in Cancer (GDSC)	Offering drug response data and genomic markers of sensitivity.	https://www.cancerrxgene.org/	[29]

Genotype-Tissue Expression (GTEx) project	Characterizing gene expression and regulation in many human tissues and associated with genetic variation and disease.	https://commonfund.nih.gov/GTEx/	30
The International Human Epigenetics Consortium (IHEC)	Generating and releasing reference epigenome maps of normal and disease tissues.	https://epigenomesportal.ca/ihec/	31
The Universal Protein Resource (UniProt)	Comprehensive resource of protein sequence and annotation data.	http://www.uniprot.org/	32

This section is organized as follows: (1) we first give an introduction about the source of batch effects, (2) The design to minimize batch effects, (3) the commonly used normalization methods, (4) the advancement of batch effects correction algorithms, (5) the progress of batch effects detection approaches, (6) the purpose and significance of the study and (7) the main contents of this study.

1.2 The source of batch effects

According to the potential source of the batch effects, which can be categorized into three stages: (1) batch effects derived from experimental design, (2) Batch effects derived from experiment processing, (3) Batch effects derived from experiment data extraction.

Batch effects derived from experimental design, which in a high-throughput omics data set, corresponding to the unnecessary experimental bias that masks the experimental results due to the inappropriate initial experimental design[33, 34]. For instance, the normal group is all from the elderly man, while the treatment group is from the young woman. Those phenotype differences, such as age or gender can mask the difference of samples[35-37]. As a result, the experimental design affects the whole data set.

Batch effects derived from experiment processing, which is related to the different protocols (e.g., feed, temperature, and platform), as well as variation from different labs or places where samples were collected or prepared, different handling personnel and reagent lots[37, 38]. For instance, in the process of collecting samples for RNA preparation, we first need to store the collected biological samples (tissues or cells) in a suitable way. Generally, we need to store the biological samples in liquid nitrogen in a timely manner. If we don't freeze tissue or cells immediately when we collect them, they may differentiate. Studies have shown that when the same tissue is extracted and placed at different times, there will be obvious changes in gene expression differences, which will affect the results of the experiment[39, 40]. In addition, RNA from samples that are not immediately frozen may be degraded by RNase. This degradation can have a significant effect on experimental results[41]. Furthermore, in a large cohort study, sample preparation can cost several weeks and include different

technicians and reagent lots[42].

Batch effects derived from experiment data extraction, which is associated with array handling effects (e.g., storage, fixing, and washing), scanner effects, personnel effects, software effects, calculation methods and parameter selection in pipelines[37, 42]. For instance, in the microarray image analysis, different analysis approaches and software can produce different results[34]. Therefore, it's important to note that all the data should be treated equally.

Those factors mentioned above might generate variation not associated with the biological interest, which we need taken into account. If the batch effects are not handled properly, the batch effect may produce false positives and false negatives[5, 17]. In severe cases, the experimental conclusions will be invalid or not reproducible[43]. In general, reproducibility can be improved through rigorous experimental protocols, but using rigorous experimental protocols can not completely solve the batch effects problem. Batch effects, however, can be alleviated through experimental design and batch effect correction approaches. Here, we first introduce the general recommendations for the experiment design and then describe approaches that can be used for batch effect correction[44].

1.3 Designing to minimize batch effects

The experiment design plays a key role in helping to minimize the batch effects, and it can critically affect the result[45]. The purpose of experimental design is to make experiments more economical and efficient, to verify our hypotheses reasonably, and to optimize the content of experimental data by accounting for the different sources of variation (e.g., experimental variation, biological variation, and technical variation) at the experimental plan level[44]. A good experimental design allows us to obtain the biological signals that we are interested in, and at the same time, it can reduce experimental errors and technical errors[46]. Reversely, a bad experimental design will not only cover up the biological signals but also make the data unusable, leading to false discovery[44, 47, 48].

To ensure reproducible results, we need to consider not only biological factors but also technical factors. For example, record RNA extraction batches, protein digestion batches, as well as the precise values of the room's temperature and

humidity, which helps to track down the problem. At the same time, when the instrument processes data, it is usually performed batch by batch (due to the limited number of samples that the instrument can process at one time). This information also needs to be recorded in detail. Recording batch information will help us choose the appropriate batch effects correction algorithm later. In summary, during the initial experimental design process, we can prevent real biological signals from being confused by understanding these factors that might affect the sample.

We should avoid complete confound between biological states and technical factors since in this case, the biological signals are completely hidden. For example, the normal group is collected in one hospital, and the disease (e.g., tumor) group is collected in another hospital. Even a gene is detected being a significant gene, we do not know it was due to the class effects or batch effects. This should be avoided at the beginning of the experimental design. A common way to minimize this problem is by blocking[46]. Blocking try to make each batch containing an equal number of samples from each group[46]. In summary, balancing experimental groups (normal and cancer groups) in each batch can make batch effects to be corrected in subsequent analyses[44, 46].

In the experimental stage, another critical factor that needs to be considered is replication[49-51]. We should not only do biological replication to reduce the impact of biological variation between samples, but also need to include technical replication to reduce the impact of technical variation (batch effects)[51-53]. There should be note that, adding replication will increase costs. To address the trade-off between research cost and biological signal detection sensitivity, statistical power is needed[52]. Statistical power can help us to determine the need for biological and technical replication to gain the desired sensitivity. There are several methods that exist for the statistical power estimates[44, 54, 55]. In summary, choosing appropriate biological and technical repetitions based on statistical power has a significant impact on mitigating batch effects and obtaining robust biological signals.

1.4 Normalization used as a proxy procedure to correct batch effects but with limited efficacy

Normalization is a procedure for adjusting the global properties for each sample so that they may be cross-compared[12, 15]. Popular normalization methods include linear scaling, quantile normalization, and Z normalization. Normalization is at times used as a proxy procedure for the removal of batch effects. However, with limited efficacy as the nature of batch effect is complex and may not impact all variables similarly. This has been demonstrated repeatedly: Luo et al.[15] and Leek et al[17] both pointed out batch effects persist post-normalization in their microarray datasets. This finding is platform-independent as it is also consistent in proteomics datasets[12]. In addition, different normalization methods have strong influence on the results of gene identification[56]. Normalization approaches are useful for aligning samples such that they have similar properties, which facilitates the use of statistical feature-selection algorithms. However, they are not canonical batch effect correction algorithms and we do not consider them as such[5, 12].

Although there are novel studies shown that specific normalization methods (e.g., Gene Fuzzy Scoring or GFS[57] together with network-based analysis techniques[12, 58]) are batch effect resistant and without affect data integrity, these methods are imperfect either. Both do not directly estimate and correct batch variation from the data, and also introduce further limitations, for example, GFS only considers genes/proteins with higher abundance, while the network only evaluates genes/proteins that overlap with it. Therefore, algorithms that estimate and correct batch effects are highly necessary, and these approaches had been developed and known as batch effect correction algorithms.

1.5 Batch effects correction algorithms (BECAs)

There are various BECAs exist, according to the principle used, BECAs can be categorized into three main types: (1) location-scale based methods, (2) matrix factorization based methods, and (3) hybridization methods. We have summarized the commonly-used BECAs and their major advantages and disadvantages in Table 1-2.

Table 1-2. Summary of representative batch effects correction algorithms (BECAs)

BECAs	Types	Advantages	Disadvantages	Reference
DWD	Location-scale based methods	Improves statistical power when batch-class design is balanced	1. Complexity 2. Not robust when sample sizes of data are small 3. Not robust to outliers 4. Can only be applied to two batches at a time 5. Potential biological effects may also be removed 6. Batch factors must be known	[13]
BMC	Location-scale based methods	1. Improves statistical power when batch-class design is balanced 2. Simple and fast	1. Not robust when sample sizes of data are small 2. Not robust against outliers 3. Potential biological effects may also be removed 4. Batch factors must be known	[59]
Ratio-A	Location-scale based methods	1. Improves statistical power when batch-class design is balanced	1. Not robust when sample sizes of data are small	[15]

		2. Simple and fast	2. Not robust against outliers 3. Potential biological effects may also be removed 4. Batch factors must be known	
Ratio-G	Location-scale based methods	1. Improves statistical power when batch-class design is balanced 2. Simple and fast 3. Robust against outliers	1. Not robust when sample sizes of data are small 2. Potential biological effects may also be removed 3. Batch factors must be known	15
ComBat	Location-scale based methods (Note: Improved location-scale methods by including the Empirical Bayes approach)	1. Improves statistical power when batch-class design is balanced 2. Robust against small sample size 3. Robust against outliers 4. Generally fast	1. Potential biological effects may also be removed 2. Batch factors must be known	14
SVA	Matrix factorization based methods	1. Improved statistical power when batch-class design is balanced 2. Estimates unknown batch effects without need batch factors	1. Complexity 2. Not robust against outliers 3. Potential biological effects may also be removed	60

RUV	Matrix factorization based methods	1. Improves statistical power when batch-class design is balanced 2. Estimates unknown batch effects without need batch factors	1. Complexity 2. Not robust against outliers 3. The reference probe can be affected by several reasons 4. Potential biological effects may also be removed	61
Harman	Matrix factorization based methods	1.Improves statistical power when batch-class design is balanced 2.User guided 3. Flexible	1. Complexity 2. Not robust against outliers 3. Batch factors must be known	62
FA-batch	Hybrid methods	1. Improves statistical power when batch-class design is balanced 2. Removes batch effects from both training and validation data, good for the prediction task. 2. Robust against outliers	1. Complexity 2. Only applicable in the presence of a binary target variable 3. Not suitable when the batch effects are weaker than the class effects 4. Batch factors must be known	63

1.5.1 Location-scale based methods

Methods in this category convert the data from each batch with similar mean and/ or variance of each feature (such as gene or protein)[16]. They assume these transformations can make the data more comparable and without deleting any useful biological signals[16].

Well known BECAs in this category include distance-weighted discrimination (DWD)[13], batch mean-centering (BMC)[64], ratio-based methods[15], and an Empirical Bayes method, called Combating Batch Effects (ComBat)[14].

In DWD, it was based on the support vector machines (SVM)[13]. It first finds the best hyper-plane to separate the samples from two batches, then the sample in each batch are projected on the DWD plane, the batch average was found, and then subtract the DWD plane multiplied by the average[13]. DWD can improve the statistical power when batch-class design is balanced. However, DWD is complex to use, sensitive to outliers and sample size, may remove potential biological effects and it needs the batch information known *a priori*. Also, when there are multiple batches (>2), it is less effective since it can only take account of two batches at a time[14].

In BMC, it converts the data by doing batch-wise subtracting of the variables by their arithmetic means, therefore, the mean of each gene becomes zero[64]. BMC is simple to use and work fast. However, BMC may remove potential biological effects, and it is sensitive to the outliers and sample size of the data.

In ratio-based methods, there are two types that existed. One is the ratio-based method by using the arithmetic mean as reference (Ratio-A), the other is the ratio-based method by using the geometric mean as reference (Ratio-G). In Ratio-A, it transforms the data by doing batch-wise dividing of the variables by their arithmetic means[15]. In Ratio-G, it converts the data by doing batch-wise dividing of the variables by their geometric means[15]. Ratio-A and Ratio-G are easy to use and work fast, in addition, Ratio-G is resistant to the outliers if there are exist in the data. However, Ratio-A is sensitive to the outliers.

In ComBat, it first standardizing the data to let all the genes with similar mean and variance, then the empirical Bayes method is applied to the standardized data to estimate the presence of batch effects[14]. The original data then corrected with the

calculated batch effect estimator. ComBat is robust against the small sample size and outliers of the data. However, ComBat may also remove potential biological effects and need to know the batch factors *a priori*. On the basis of ComBat, some improved algorithms are proposed for different scenarios. Stein et al.[65] proposed the modified ComBat used for the case where one batch in the training data. Zhang et al.[66] proposed the mean-only ComBat used for the situation where the batch effects is not severe, and reference ComBat used for the situation where there is a high-quality batch can used for reference. Zhu et al.[67] developed an autoComBat used for automatically determining of whether to use the parametric or nonparametric version of ComBat.

In general, the location-scale based methods reliance on the known batch factors, when that information is limited because of incorrect labeling, sample exchange, and personnel changes which commonly existed in the big project (e.g., The Cancer Genome Atlas (TCGA)[26], Encyclopedia of DNA Elements (ENCODE)[68], Genotype-Tissue Expression (GTEx) project[69] etc), these methods are no longer feasible. Some Matrix factorization based methods were introduced to overcome this limitation by estimating the batch effects from the full data matrix[5, 70], the representative methods are surrogate variable analysis (SVA)[60] and remove unwanted variation (RUV)[61]. Besides, there are also exist some other matrix factorization methods that try to overcome the overcorrection problem by setting a threshold, the representative method is the Harman. We described below:

1.5.2 Matrix factorization based methods

Methods in this category are based on the matrix factorization methods (e.g., singular value decomposition (SVD)[71] or principal component analysis (PCA)[72, 73]). The commonly used batch effects correction methods in this category include SVA[60], RUV[61], and Harman[62].

In SVA[60], it first needs to assign the class factor and assumes that the variation not related to biological signal is possibly caused by batch effects. After estimated the batch associated variation, SVD is deployed to remove those batch effects[60]. SVA is a useful method, but it is complicated to use[35], and it is also sensitive to class information. When the supplied class information is wrong owing to mislabels, SVA can make errors[60, 74]. Based on SVA, there are some improved algorithms applied in

different situations. Parker et al.[75] developed permuted SVA (pSVA) which can avoid algorithms to remove unknown biological difference signals. Leek et al.[76] proposed supervised SVA (SSVA) for the genomics data when there are control genes exist, and SVASEQ for the RNA-seq data by doing a log transformation before the standard SVA approach. Chakraborty et al.[77] introduced the SVAPLSseq for the RNA-seq data by using a Partial Least Squares (PLS) algorithm to estimate and correct the batch effects. The author showed that the advantage of SVAPLSseq is more flexible and extensive compare with the existing SVA method[77]. Parker et al.[78] developed the frozen surrogate variable analysis (fSVA) used for the prediction problem.

In RUV[61], it requires negative control genes (either spike-in controls or housekeeping genes) *in priori*. Those genes are regarded as uncorrelated with class effects, and can be used to estimate the batch effects. Although this method is useful, it is sensitive to the quality of the control genes, and also some control genes are reported to be associated with diseases. This method was first introduced by Gagnon-Bartsch et al.[61] used for the batch effects in the microarray data. There are several extensions based on RUV developed. Risso et al.[79] developed the RUVg, RUVs, and RUVr for the batch effects correction in the RNA-seq data. RUVg by using the negative control genes, RUVs by using the negative control samples, and RUVr by using the residuals. Those genes, samples, and residuals are assumed to be not associated with class effects[79].

Harman[62], a method based on the PCA approach, but required both class factor and batch factor known *a priori*[62]. The method first transforms the data into its PC, and then it scans each PC to find which PC is strongly associated with batch effects, followed by batch effect correction with a defined threshold (commonly at 95%) in order to preserve the class effects. The threshold 95% means only 5% class effects will be removed when applying Harman to do batch effects correction[62]. The advantage of this method is maintaining a good trade-off between the class effects protection and batch effects correction by a user-defined threshold. While the disadvantage of this method is that when the class and batch information is unknown or mislabeling, it will not appropriate.

1.5.3 Hybridization methods

Hornung et al.[63] proposed a hybrid approach called factor adjustment batch

(FAbatch), which is a method integrating the location-scale and matrix factorization approach together. The author demonstrated that this method is competitive in many situations compare with other methods (ComBat, SVA, Ratio-A, Ratio-G, BMC)[80]. The defect of this method is that it fails to protect enough class effects when the batch effects are weaker than the class effects.

In practice, knowing the class and batch factors in advance is difficult, especially in a big project (e.g. TCGA [26]), due to mislabeling, personality changes. Although the batch factors may be estimated, it still difficult to distinguish the biological signal of interest and technical heterogeneity. And removing batch effects may have an adverse impact on the data integrity[5, 81].

1.6 Batch effects detection approaches

Batch effects detection is a key step in evaluating the performance of batch effects correction. In general, batch effects detection is done by visualizing and/or computing certain performance indices between the raw data and corrected data. In this part, we describe different approaches for batch effects detection. Commonly, batch effects detection approaches can be divided into two main categories: qualitative analysis tools and quantitative analysis tools. Fig.1-3 shows the different batch effects detection approaches.

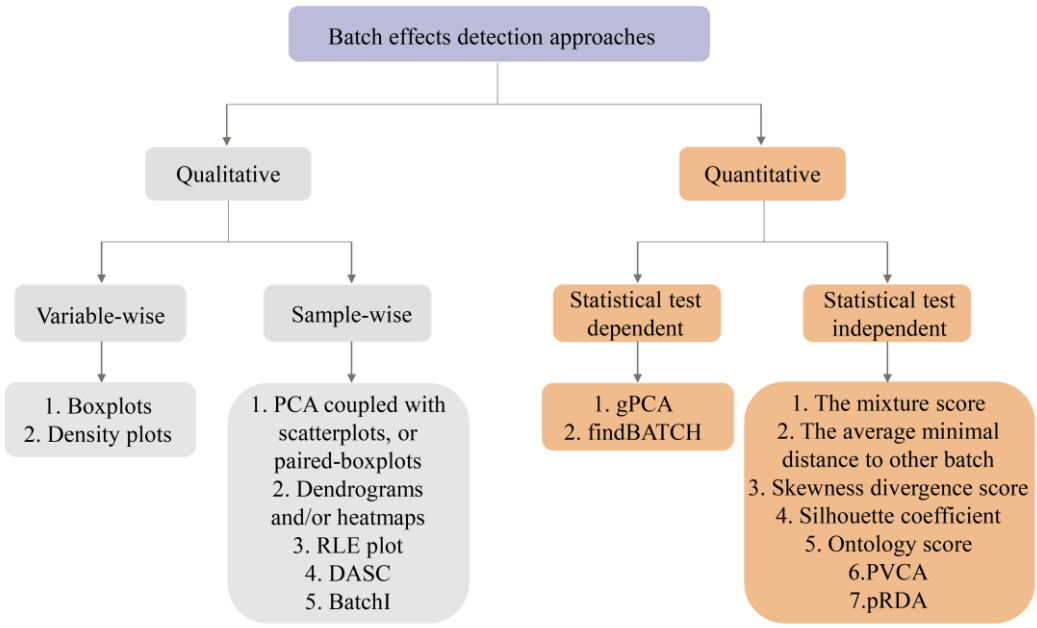


Fig. 1-3. Classification of batch effects detection approaches.

1.6.1 Qualitative analysis approaches

Methods in this category using a direct way to evaluate the batch effects by visualization means. The commonly used qualitative analysis tools in this category include boxplots[16], density plot[16], dendrograms[16], PCA (e.g. plot 2 dimensional PCA scatterplot, principal component paired boxplots[12]), relative log expression (RLE) plots[82], data-adaptive shrinkage and clustering (DASC)[70], and BatchI[83].

In the boxplots of gene expression distribution (variable wise). It assumes that the expression level of the same variable (e.g., gene) will have a similar distribution in two different studies if there is no batch effect. And the density plot employs a similar idea[16]. They just offer a local view of batch effects, while for a global view of batch effects, plotting at sample level is necessary.

Hierarchical clustering, often presented in a dendrogram, assumes that those samples derived from the same category will group together regardless of whether they are from different studies or not, if batch effects do not exist[16].

RLE, a method used to detect the unwanted variation caused by batch effects, it assumes that most variables are not affected by class effects, therefore, if there are exist sample heterogeneity (i.e., different distributions, different medians) may suggest the existence of batch effects[82].

PCA, which is a statistical method that aims for complexity reduction through transform high-dimensional data into a few numbers of principal components (PCs), these PCs are linearly uncorrelated variables[73]. The PCs are ranked by the contribution to the total variance. Given the PCs, we can check for the presence of batch effects with scatterplots or paired boxplots.

DASC, which can be used to detect batch effects without the prior knowledge of batch factors[70]. It was dependent on the data-adaptive shrinkage (DAS)[84] and semi-Non-negative Matrix Factorization (semi-NMF)[85]. The first step of this method is to apply DAS to the data to get a batch matrix, and then the batch factors were extracted from the batch matrix by the semi-NMF method. A dendrogram plus the heatmap were used to show the batch effects.

BatchI, which can be used to detect batch effects without the prior knowledge of batch factors[83]. It first divides the high-throughput data into sub-series associated with the estimated batches, and then uses the method of dynamic programming[86] to split the data under the condition of maximum dispersion between batches but keeping the

minimum within batch dispersion[83].

Qualitative analysis tools are commonly used as the first quick check on the results and provide a rough estimate of the effectiveness of batch effect correction, however, they suffer subjective interpretation. In order to conduct a more objective evaluation, quantitative method should also be used to evaluate the quality of batch effect correction.

1.6.2 Quantitative analysis approaches

The commonly used batch effects correction methods in this category include two types: statistical test dependent and independent. For the former, there are guided-PCA (gPCA)[87], and findBATCH[88]. For the latter, there are the mixture score[89], average minimal distance to other batch[63], the skewness divergence score[63], silhouette coefficient[90], and ontology score[91], principal variance component analysis (PVCA)[42], partial redundancy analysis (pRDA)[37, 92].

gPCA, an extension method of PCA, it design to overcome the problem that when the batch effect is not the largest source of variation in the tested data, the traditional PCA method is not very effective[87]. gPCA using input batch factors, it guides the SVD to find the batch effects that existed in the data. In the gPCA, there are two metrics, gPCA delta and *P*-value, gPCA delta quantifies the variance associated with batch effects, and the *P*-value offers support for the confidence of estimated gPCA delta[87].

findBATCH, a method based on probabilistic principal component and covariates analysis (PPCCA) to detect the batch effects[93]. The main difference between gPCA and findBATCH is that the statistical test of gPCA is based on global PCs, while the statistical test of findBATCH is based on each PC[93].

The gPCA and findBATCH are fast and simple to use. However, they do not provide information related to class effects. Also, the gPCA *P*-value can not conveys confidence about the absence of batch effects[94].

Beside statistical test dependent quantitative analysis tools, there are also exist statistical test independent quantitative analysis tools.

The mixture score, which evaluate how samples from different batches are mixed by employing the idea of k-nearest neighbor based distance[89, 95]. The mixture score is ranged from 0-1, a value close to extremities suggests that the batch effects

are strong, while a value nearly to 0.5 indicates that batch effects had been removed or absented.

The average minimal distance to other batch, which assesses the distance of different batches based on the Euclidean distances. A smaller value after batch effects correction suggests that batch effects had been corrected[80].

The Skewness divergence score, which assesses distribution differences between different batches of data, a smaller value after batch effects correction implies that batch effects had been adjusted[63].

The silhouette coefficient, which has been used to evaluate the consistency of the sample group according to the class or batch[90]. Its value ranges from -1 to 1, a value close to 1 indicates samples are correctly grouped. While a value close to -1 indicates samples are wrongly grouped.

The ontology score, which based on the *Cell Ontology* to evaluate if batch correction results in a better consistency between pairwise similarity that observed and cell type similarity inferred from ontology[91].

Although these metrics offer conceptual information regard to the quality of batch effects correction, they do not provide class effects information. While this defect can be overcome by PVCA and pRDA.

PVCA, which is based on the principal component analysis (PCA)[72] and variance component analysis (VCA)[96, 97]. It first applies the PCA into the high-dimensional data for the dimensionality reduction, then estimating the proportion of variation associated with batch effects based on VCA[42, 98]. Although this approach has proven to be effective in detecting batch effects and class effects[99-103], it suffers from the following limitations: (1) the statistical power may be reduced since this approach includes several steps, (2) standard practice for choosing the best numbers of PCs related to the data is lacking, and (3) batch factors need to know in advance[93].

pRDA, which requires both class and batch factors, it can estimate the variation associated with class and batch effects based on multivariate linear regression and PCA approach[37, 92]. This method is demonstrated as powerful and convenient[37]. However, in some projects, only a few potential sources of batch were recorded.

1.7 Purpose and significance of this study

Batch effects can cause false effects or mask real effects in high-throughput data study, this in turn, impedes the development of biomarker and drug target identification in the medicinal application. A wide variety of batch effect-correction algorithms (BECAs) are designed to handle this problem, and there are various comparative evaluations have been conducted to establish the best BECA. We hold the viewpoint that the notion of best is context-dependent. Moreover, alternative questions beyond the simplistic notion of "best" are also interesting: Are BECAs robust against various degrees of confounding and if so, what is the limit?

The second focus of my work is the development of new methods for batch effect correction based on the exist methods. ComBat has been popularly used for batch effects correction, especially under the small sample sizes scenarios. However, when deploying ComBat to do batch-effect correction, only the batch information is specified but not class information. In the case of batch and class effects are confounded, we suspect that this can lead to performance issues with regards to proper batch effect inference and removal. To solving this gap, we propose an alternative procedure to ComBat, the class-specific ComBat (CS-ComBat), where batch effects on each sample class are corrected independently before being merged.

The third focus of my work is on investigating the performance of batch effects correction under the peptide level and the protein level of proteomics data. At present, batch effects correction is done at the protein level to remove batch effects. However, batch effects correction done at the earlier peptide level has not been investigated. We suspect that if the batch correction is done earlier at the peptide level, would the protein level be more accurate.

Based on the above investigations, this study would (1) help users of BECAs become more aware of its limitations, and therefore, can be more strategic in its application. (2) Provide new and potential effective batch effects correction methods to this filed; (3) Allow researchers to pay more attention to the batch effects issue and alleviate the impaction of batch effects to their experiment; (4) Provide a better understanding of the complex biological mechanisms that cause disease, developing improved diagnostics, prognostics, and therapeutics and for offering potential drug targets.

1.8 Main contents of this study

The main contents of this study were as following:

(1) Examining the practical limits of batch effect-correction algorithms: When should you care about batch effects?

(2) More robust batch correction with ComBat in data with batch-class confounding issues.

(3) Class specific-ComBat for correcting batch effects in high-throughput data with a focus on small sample size.

(4) Batch effect correction on proteomics data: correcting at peptide level or protein level?

Chapter 2. Examining the practical limits of batch effect-correction algorithms: When should you care about batch effects?

2.1 Introduction

Batch effects are technical sources of variation arising from different experiment times, handlers, reagents, and instruments[5, 17, 74]. Batch effects are especially relevant to assays with sample-size limits, thereby requiring multiple runs (each run constituting a "technical batch"). Batch effects are not a homogeneous group of technical variation: Specific assays and platforms can create idiosyncratic batch effects such that within-batch effects and between-batch effects[104, 105]. In this study, we do not cover these types of batch effects as it is very specific, instead we simulated batch effects using naïve but more generalizable assumptions. Batch effects may confound functional analysis. In a comparative analysis where one phenotype class is compared against another (e.g., normal against disease), Batch effects may induce false effects (false positives) or mask real effects (false negatives)[81]. These issues hamper the proper development of diagnostic/prognostic markers. Thus, it is important that batch effects are dealt with satisfactorily.

There is a wide variety of batch effect-correction algorithms (BECAs), necessitating various comparative evaluations[15, 16, 99]. Although ComBat[14], often emerges as the top performer, other methods such as Harman[62], surrogate variable analysis (SVA)[60], the ratio-based algorithms[15], Ratio-A, Ratio-G as well as Batch mean-centering (BMC)[64], have also been reported to work well in their own specific evaluations. Due to such disparities, we are curious whether the outcomes of such ranking exercises are highly context-dependent (e.g., due to the data used or upstream transformation).

Batch effect-correction is the process of estimating and removing batch effects algorithmically. There are several potential factors influencing batch effect-correction: (1) the degree of confounding between class and batch factors, (2) upstream

normalization procedure, and (3) the nature (magnitude and variability) of the batch effect itself.

Confounding refers to the extent class and batch effects are inter-mixed and indistinguishable. Suppose we have two classes, A and B, 4 samples per class. Amongst 8 samples, the order of the class labels S is (A_1, A_2, A_3, A_4, B_5, B_6, B_7, and B_8). This ordered arrangement of class labels, A and B, denotes the assignment of each sample to a particular class. In biology, a class can be a phenotype, e.g., disease or normal. The list S, being an ordered assignment of class labels, is also referred to as the class factor. The corresponding batch factor, X, given two batches, U and V, may be assigned as (U_1, U_2, V_3, V_4, U_5, U_6, V_7, and V_8). Here, although batch effects do exist, the batch factors are equally distributed between the samples in classes A and B. This is a 'balanced scenario'. In experimental design, the 'balanced scenario' is preferred as it is expected that, given sufficient sample size and random distribution, Batch effects and other sources of technical bias may be "averaged out". Since Batch effects are negated, there is no confounding between class and batch factors.

Unfortunately, experimental designs are often inevitably imperfect. Real-world factors, e.g., sample availability and failed experiments may also be contributors. These lead towards batch-class factor imbalance, such that Batch effects cannot be negated via "averaging out". For example, given (A_1, A_2, A_3, A_4, B_5, B_6, B_7, and B_8), we may have (U_1, U_2, U_3, V_4, U_5, V_6, V_7, and V_8) instead. Class A is over-represented in batch U (75%), and *vice versa*. Here, if a strong difference is detected between samples in class A against B, the differential effect may be contributed by both class and batch effects. It is hard to tell apart which the true source(s) of the differential effect is; we therefore say that class A is confounded with batch U.

In the most extreme scenario, given (A_1, A_2, A_3, A_4, B_5, B_6, B_7, and B_8), we may have (U_1, U_2, U_3, U_4, V_5, V_6, V_7, and V_8). This situation may occur due to complete oversight in experimental design. Here, class A is completely correlated with batch U (100%), and *vice versa*. If a strong difference is detected between samples in class A against B, there is no way to tell if the difference is due to true class effects or batch effects. We therefore say that class A is perfectly confounded with batch U.

BECAs are techniques for estimating and removing batch effects from data. Each BECA makes assumptions about the underlying data distribution and is theoretically susceptible to upstream normalization procedure. Appropriate

normalizations are meant for unwanted technical variation. However, normalization may change the scale and distribution of the data, and consequently, impact BECA performance.

Determining the nature of batch effects is a challenging issue: A simple perspective is to assume that batch-effect correlated variation is uniformly distributed across all variables (variables being measurable quantities that vary between different samples, e.g., gene or protein expression). Leek et al.[17] suggested that batch effects are non-homogeneous, and non-uniformly distributed amongst different variables. If Leek et al. are correct, then most batch-effect simulation approaches assuming uniformity would be insufficient. Moreover, it is probable that uniformity assumptions are rather unrealistic and can be easily dealt with analytically[12].

This work is not a ranking exercise for elucidating the best BECA given some dataset(s). Instead, we are interested in examining the practical limits of BECAs given (1) different upstream normalization procedures, (2) if generic normalization methods can actually deal with batch effects at all, (3) where there are various levels of confounding, which BECAs remain effective, and finally, (4) is there reason to believe a universal best BECA exists given any scenario, or whether data-dependent contexts are crucial in determining performance. To make a case for generalizability, 3 representative RNA-Seq and 1 proteomics dataset are used for evaluation.

2.2 Materials and methods

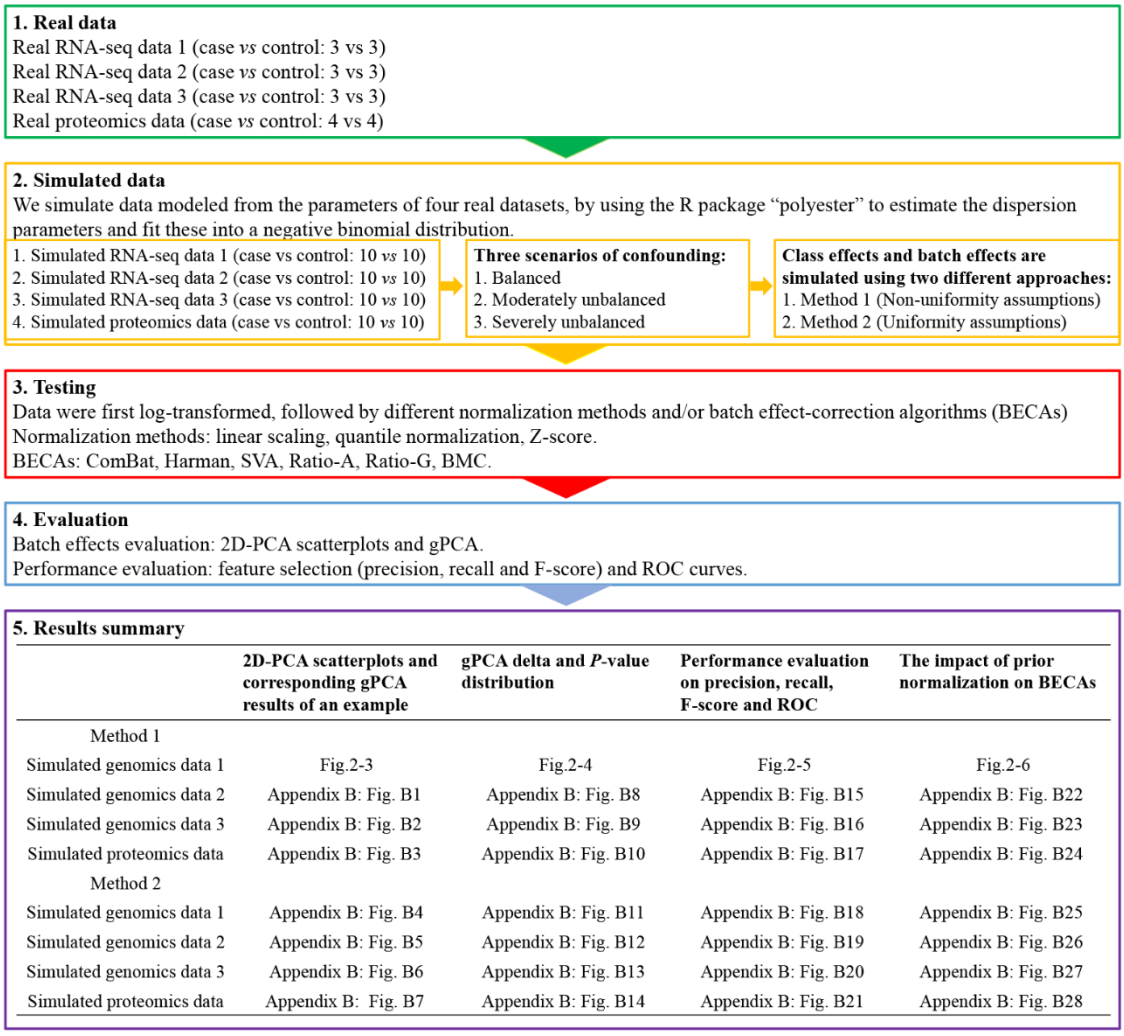
2.2.1 Simulated datasets based on real datasets

We simulated data modeled from the parameters of four real datasets: three RNA-Seq datasets, each dataset with three pairs of case and control samples[106-108], and an unlabelled shotgun proteomic dataset with four cases and controls[109]. The summary information of these datasets is shown in Table 2-1. An overview is provided in Fig. 2-1 linking datasets to the respective analytical output and corresponding figures as referenced.

Table 2-1. The data used in this study

Data	Accession Number or download link	Sample size of real data	Phenotype classes of real data (control vs case)	Sample size of simulated data based on real data	Phenotype classes of simulated data (control vs case)	Omics Platform	Depth (number of reads per sample) or acquisition mode	Reference
RNA-seq data 1	GSE53334	6	3 *vs* 3	20	10 *vs* 10	Illumina HiSeq 2000	~50 million	[106]
RNA-seq data 2	GSE122138	6	3 *vs* 3	20	10 *vs* 10	Illumina HiSeq 2500	15~20 million	[107]
RNA-seq data 3	GSE106417	6	3 *vs* 3	20	10 *vs* 10	Illumina HiSeq 2500	5~25 million	[108]
Proteomics data	Download link[a]	8	4 *vs* 4	20	10 *vs* 10	Label-free shotgun proteomics LC-MS/MS	Data-dependent acquisition (DDA)	[109]

Download link[a]: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2830842/bin/supp_M900260-MCP200_TableS3_total_result.xls



	2D-PCA scatterplots and corresponding gPCA results of an example	gPCA delta and *P*-value distribution	Performance evaluation on precision, recall, F-score and ROC	The impact of prior normalization on BECAs
Method 1				
Simulated genomics data 1	Fig.2-3	Fig.2-4	Fig.2-5	Fig.2-6
Simulated genomics data 2	Appendix B: Fig. B1	Appendix B: Fig. B8	Appendix B: Fig. B15	Appendix B: Fig. B22
Simulated genomics data 3	Appendix B: Fig. B2	Appendix B: Fig. B9	Appendix B: Fig. B16	Appendix B: Fig. B23
Simulated proteomics data	Appendix B: Fig. B3	Appendix B: Fig. B10	Appendix B: Fig. B17	Appendix B: Fig. B24
Method 2				
Simulated genomics data 1	Appendix B: Fig. B4	Appendix B: Fig. B11	Appendix B: Fig. B18	Appendix B: Fig. B25
Simulated genomics data 2	Appendix B: Fig. B5	Appendix B: Fig. B12	Appendix B: Fig. B19	Appendix B: Fig. B26
Simulated genomics data 3	Appendix B: Fig. B6	Appendix B: Fig. B13	Appendix B: Fig. B20	Appendix B: Fig. B27
Simulated proteomics data	Appendix B: Fig. B7	Appendix B: Fig. B14	Appendix B: Fig. B21	Appendix B: Fig. B28

Fig. 2-1. Overview of simulation pipeline from dataset to results.

We used the R package "polyester" to estimate the dispersion parameters of both RNA-Seq and proteomics samples, and fitted these into a negative binomial distribution[110]. To demonstrate a good fit between simulated and real data, we used the mean/standard deviation plots (Fig. 2-2).

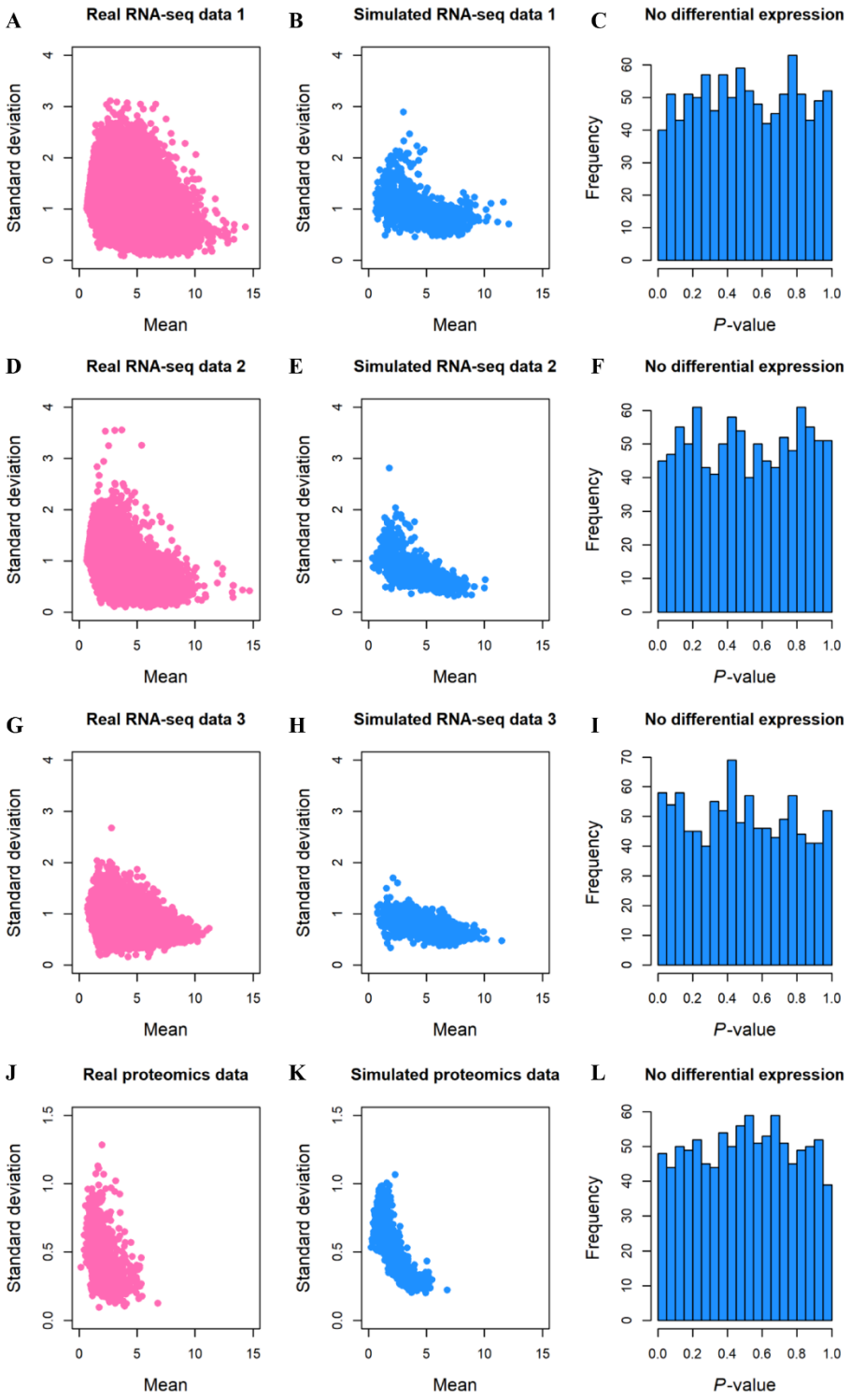


Fig. 2-2. Mean and standard deviation distribution of real RNA-seq and proteomics data and simulated RNA-seq and proteomics data (**A**, **B**, **D**, **E**, **G**, **H**, **J and K; c.f. Table 1**). *P*-value plot shows the uniform distribution for the simulated RNA-seq and proteomics data without addition of class and batch effects (**C**, **F, I and L**).

We simulated 100 expression studies for RNA-seq and proteomics datasets, respectively. For each study, we simulated expression for 1000 genes or proteins on 20 samples with no class effects and batch effects. We used a histogram showing the distribution of *P*-values to confirm that simulated data without added effects did not present abnormally high false positives, beyond what is expected by chance (Fig. 2-2).

In this study, we simulated three scenarios of confounding: (1) balanced batch-class ($R^2 = 0.0$), (2) moderately unbalanced batch-class ($R^2 = 0.4$), (3) severely unbalanced batch-class ($R^2 = 0.8$). The R^2 is normally for explaining fit in linear regression. Here, it is used for indicating confounding between class and batch[76]. Where R^2 is 0.8, it means there is 80% confounding between class and batch factors.

Class effects and batch effects were simulated using two different approaches (method 1 being the default approach, and method 2 only being used to check for corroboration):

2.2.1.1 Method 1 (non-uniformity assumptions)

For method 1, class effects and batch effects are incorporated based on the method of Leek[76]. For each simulated study (RNA-seq and proteomics), 20 samples are split into two classes (10 samples in each class). For the genes/proteins, the class factors (class A and class B) were drawn from a normal distribution with mean 0 and standard deviation 1 and simulated on 300 genes/proteins (genes 701–1000) using the R package of "polyester". Since class factors are simulated, then the differential genes/proteins are known *a priori*.

After adding class effects, the 20 samples are assigned to 2 technical batches. The batch allocation is according to the batch-class design (balanced, moderately unbalanced or severely unbalanced), batch factors were also drawn from a normal distribution with mean 0 and standard deviation 1 and simulated on 400 genes (genes 601–1000) using R package of "polyester". Because batch effects are simulated only on a subset of genes, this is referred to as a non-uniform approach.

2.2.1.2 Method 2 (uniformity assumptions)

For method 2, class-effect sizes are inserted based on the method of Langley and Mayr to distinguish control and case classes[111]. Samples are randomly assigned to classes A and B (10 samples per class). For each class B sample in the simulated RNA-seq /proteomics dataset, class-effect sizes are sampled randomly from five

possibilities or *p* (20%, 50%, 80%, 100%, and 200%), and inserted into randomly selected variables on samples belonging to class B, constituting 30% of all variables. This is expressed as:

$$SC_{i,j}{}' = SC_{i,j} * (1 + p)$$

where $SC_{i,j}$ and $SC_{i,j}'$ are respectively the count before and after *p* inserted from the *j*th sample of protein *i*.

Batch effects may also be simulated using an identical procedure by assigning each sample to a batch (1 and 2). The batch allocation is according to the batch-class design (balanced, moderately unbalanced or severely unbalanced). For each gene in batch 2, a batch effect is randomly drawn from *p* and inserted[12]. Because a batch effects is inserted into every gene in batch 2 samples, this is referred to as a uniform approach.

2.2.2 Data pre-processing

Prior to normalization and/or batch correction, both RNA-Seq and proteomic measurements are log-transformed to reduce scale and provide better data distribution symmetry[76].

2.2.3 Normalization methods

2.2.3.1 Linear scaling

Linear scaling bounds the value of data from 0 to 1 but does not affect the distribution. It is expressed as:

$$X_{\mathrm{i,\ 0\ to\ 1}} = \frac{X_i - X_{min}}{X_{max} - X_{min}}$$

Where X_i, $X_{\min}$ and $X_{\max}$ are the variable, the minimum value and the maximum value of each sample, respectively.

2.2.3.2 Quantile normalization

Quantile normalization is used to force-set samples from different data source, class or batch to identical distributions. It involves first ranking the genes/proteins of each sample by magnitude. Then, the means across same-ranked genes/proteins are calculated. All same-ranked genes/proteins are now populated with this row-mean

value. Finally, each sample is reordered back to its original configuration prior to the ranking.

2.2.3.3 Z-score transformation

Z-score transformation fits each sample to an ideal Z-distribution with a mean of 0 and standard deviation of 1. Because all samples are force fitted to the Z-distribution, this makes them in a sense, comparable. Parametric tests are now also applicable on such Z-normalized data. The Z-score, for each gene/protein, per sample is calculated by:

$$z = \frac{x - \mu}{\sigma}$$

Where x, μ, and σ are the original value of the gene/protein, sample mean and sample standard deviation, respectively.

2.2.4 Batch effects correction algorithms (BECAs)

2.2.4.1 ComBat

ComBat is a widely used BECA. It incorporates empirical Bayes frameworks based on a location-scale model for correcting the data with batch effects. ComBat works well on small sample sizes and reportedly is robust against outliers[14, 99]. The version used here is the implementation from the "sva" R package[112].

2.2.4.2 Harman

Harman is a method based on PCA[62]. Firstly, the adjustment data are separated into its principal components (PCs). Harman scans each PC to identify variance correlated with batch effects, followed by removing the batch effects given a user-defined threshold (usually at 95%) in favor of protecting the class effects, i.e., 5% class effects will be lost due to batch effects correction with Harman at threshold 95%. The version used here is the implementation from the "Harman" R package[62].

2.2.4.3 Surrogate variable analysis (SVA)

SVA is a method based on matrix factorization. It assumes the sources of variation unrelated with the class factors are associated with batch factors[60, 76], and may be removed, thereby preserving only class effects. The version used here is the implementation from the "sva" R package[112].

2.2.4.4 Ratio-based methods (Ratio-A and Ratio-G)

Ratio-based methods scale the expression values by the arithmetic or geometric means of a group of reference samples in each batch. Ratio-A and Ratio-G are used to represent the ratio-based approaches using reference based on arithmetic (A) and geometric (G) means, respectively[15]. The versions used here are the implementations from the "bapred" R package[80].

2.2.4.5 Batch mean-centering (BMC)

BMC assume there are multiplicative systematic batch effects[64]. After BMC adjustment, the average value of each feature of all samples in each batch becomes zero. The version used here is the implementation from the R "bapred" package[80].

2.2.5 Batch effects detection methods

2.2.5.1 Principal components analysis (PCA)

PCA is a statistical method used for complexity reduction by collapsing high-dimensional data into lower numbers of linearly uncorrelated variables. These are known as PCs[73]. The PCs are ordered in decreasing order, based on the contribution towards total variance. Given the PCs, we may check for the existence of batch effects (and also, the strength of class effects) using 2D or 3D scatterplots.

2.2.5.2 Guided-PCA (gPCA)

PCA scatterplots are visual representations and do not provide an objective evaluation of batch effects. gPCA may be used instead for this purpose[87]. gPCA provides 2 metrics, a delta (δ) and its associative *P*-value. Both values are bound between 0 and 1. However, they are interpreted differently: A large value of δ (near 1) is indicative of strong batch effects. An associatively low *P*-value (≤0.05) provides support on the confidence of the estimated δ.

2.2.6 Performance evaluation

2.2.6.1 Statistical feature selection analysis

Statistical feature selection was performed using the unpaired *t*-test with a *P*-value cutoff at 0.05. No multiple test corrections were performed, as these would artificially boost precision while penalizing recall (resulting in an unfair evaluation). Statistically significant genes/proteins were evaluated using the metrics below.

For any given BECA, we may evaluate its performance on the simulated datasets where the true features (i.e., differential genes or proteins) are known *a priori*, using precision and recall:

$$Precision = \frac{TP}{TP + FP}; Recall = \frac{TP}{TP + FN}$$

Where *TP*, *FP*, and *FN* are the true positives, false positives and false negatives, respectively. Both precision and recall are important. The F-score (F_S) is a summary statistic used for evaluating the overall performance and is the harmonic mean of precision and recall:

$$F_s = 2 * \frac{Precision * Recall}{Precision + Recall}$$

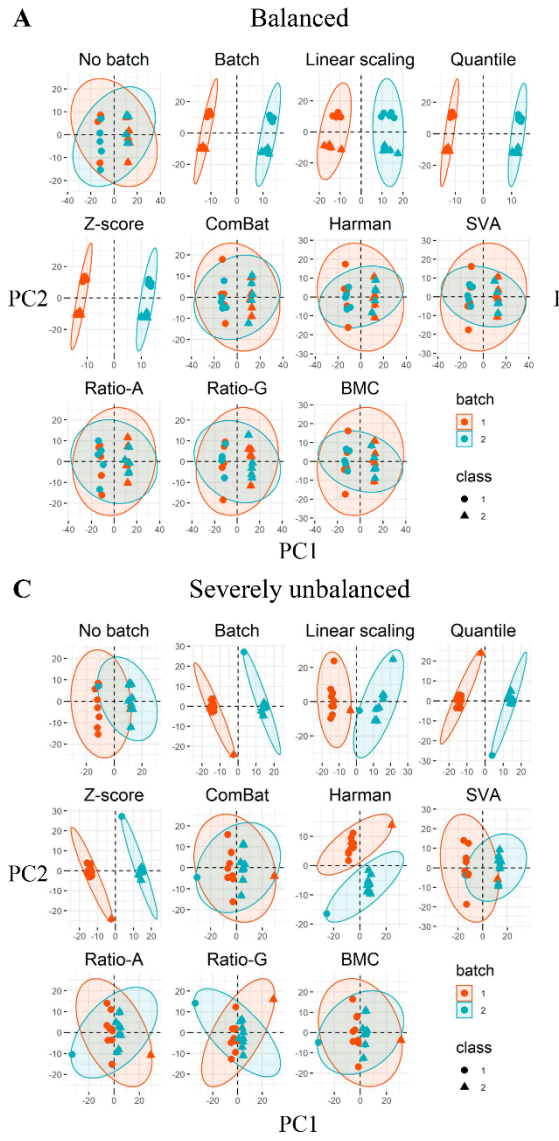
2.2.6.2 Receiver operating characteristic (ROC) curve analysis

For a given BECA applied on a simulated dataset, a ROC summarizes its overall performance in terms of sensitivity and specificity. In ROC, the area under the curve (AUC), is bound between 0 and 1. A method with good performance has large AUC (and tends towards 1). We generated 100 ROC curves per simulation. To summarize these, an averaged ROC curve was produced using the threshold-averaging approach as described by Fawcett et al[113] and used for overall performance evaluation. The ROC curves shown here were generated using the R packages "ROCR"[114] and "pROC"[115].

2.3 Results

2.3.1 BECAs are surprisingly resilient against batch-class imbalances

The 2-dimensional (2-D) scatterplot is a standard approach for detecting class and batch effects (Fig. 2-3). We will discuss findings resulted from one RNA-seq dataset based on the non-uniformity assumption in batch effects simulation (RNA-seq data 1; see Table 1 for description and meta-data of all datasets used), and then discuss if similar findings are also obtained in the other test datasets.



D

Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.17 (0.71)	0.60 (0.16)	0.93 (0.00)
Batch	0.98 (0.00)	0.95 (0.00)	0.99 (0.00)
Linear scaling	1.00 (0.00)	0.96 (0.00)	0.99 (0.00)
Quantile	0.99 (0.00)	0.95 (0.00)	0.99 (0.00)
Z-score	0.99 (0.00)	0.95 (0.00)	0.99 (0.00)
ComBat	0.05 (1.00)	0.11 (1.00)	0.26 (1.00)
Harman	0.26 (0.67)	0.37 (0.32)	0.67 (0.01)
SVA	0.16 (0.72)	0.99 (0.00)	1.00 (0.00)
Ratio-A	0.03 (1.00)	0.01 (1.00)	0.01 (1.00)
Ratio-G	0.06 (1.00)	0.08 (1.00)	0.15 (1.00)
BMC	0.01 (1.00)	0.10 (1.00)	0.03 (1.00)

Fig. 2-3. Batch-effect correction in RNA-seq data 1. **A-C**: The 2D-scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, and Z-score) and BECAs (ComBat, Harman, SVA, Ratio-A, Ratio-G, and BMC). **A**: Balanced; **B**: Moderately unbalanced; **C**: Severely unbalanced. **D**: The gPCA delta (δ) and associative *P*-values. Corresponding results for RNA-seq data 2 and 3 and proteomics data simulations are shown in Appendix B: Figs. B1 to B3. The batch-class effects are generated via the non-uniformity assumption.

In RNA-seq data 1, individual samples are resolved across PCs 1 and 2 (shapes represent class; colors represent batch). The circular boundaries are used for displaying the distributions of sample-colors, such that if a strong batch effects is present, they will not overlap.

Principal components analysis (PCA) shows that the original data cluster by class in PC1. Adding BEs leads to the formation of two batches per class in the PCA, breaking up the class-specific clustering of the original data. This effect is most dramatically observed when the color boundaries split, and is observed across all setups, from balanced (Fig. 2-3A) to severely unbalanced batch-class design (Fig. 2-3C).

As expected, when batch-class is balanced (Fig. 2-3A), BECAs perform well. BECA-adjusted data (ComBat, Harman, SVA, Ratio-A, Ratio-G, and BMC) show that samples no longer cluster by batches. However, conventional normalization approaches such as linear scaling, Z-score, and quantile do not remove batch effects.

When the batch-class design is moderately unbalanced, the BECAs still appear to work well. But not evenly so: Visual checks based on graphs can be deceptive, and although the color boundaries seem to merge, on closer inspection, many samples are still segregated by color for some BECAs. To check this better, we need a more objective measurement of batch effects. Fig. 2-3D shows the corresponding gPCA delta (and its associative *P*-value). From the moderate unbalanced scenario onwards, SVA's gPCA delta moves up dramatically (indicating batch effects are not dealt with) suggesting SVA is very sensitive to batch-class imbalance. Note that in Fig. 2-3B, PCA will regard some class effects as batch effects even in the no batch case (i.e., only class effects exist) due to confounding between class and batch. But nonetheless, class effects remain strongly correlated with PC1. As before, conventional normalization does not work in removing batch effects.

When batch-class design is severely unbalanced, distinguishing batch effects from class effects is very challenging. In Fig. 2-3C, even in the no batch scenario, the PCA plot shows samples cluster by both classes and batches in PC1 due to the high correlation of class and batch. Here, BECAs, in particular, SVA and Harman do poorly. Although the ratio-based methods (Ratio-A and Ratio-G) and BMC appear to

do well, this is deceptive, and a closer inspection reveals that the majority of samples segregate by batch. In the severely unbalanced scenario, only ComBat displays acceptable performance (although it suffers nonetheless).

Similar results are observed when considering additional datasets: RNA-seq data 2 and 3, and proteomics data (Appendix B: Figs. B1 to B3), and also when we used a different approach for simulating batch effects under the uniformity assumption in the same RNA-seq and proteomics data (Appendix B: Figs. B4 to B7). Given these simulations, the surprising observations are that: 1) BECAs are very resilient, and most are quite tolerant to batch-class imbalances; 2) in less ideal situations where there are various degrees of confounding, the ComBat approach is the best (although the Ratio-A, G and BMC methods have lower gPCA deltas, the scatterplots appear suspicious); and 3) conventional normalization cannot take care of ambient batch effects in all cases.

2.3.2 The gPCA deltas are stable but are not necessarily objective indicators of batch effects

Aside from subjective interpretability and limit in the number of observable dimensions (a 3-D scatterplot is capped at only 3 PCs max), there are also severe scalability limits in the use of scatterplots to evaluate class and batch effects. For studies involving simulations, where hundreds of simulations may be required, evaluating hundreds of graphs is neither appealing nor productive. Alternative approaches exist in the form of PCA-matrices[74] and side-by-side boxplots[12]. But these approaches require expert interpretation. A summary statistic for measuring overall batch effects per simulation is desirable and time-efficient. But does such a thing exist?

In this study, we considered the use of the gPCA delta and its associative P-value. In Fig. 2-3 (comparing Panels A–C against D), we can see that when class and batch are balanced, the gPCA delta is quite reliable. However, in unbalanced scenarios where batch effects are not yet introduced, gPCA deltas become very high. This means that the gPCA delta is very sensitive to the experiment setup and will inflate its estimation of batch effects when class-batch design is unbalanced.

The gPCA delta also has blind spots: Besides susceptibility to unbalanced design, it grossly underestimates batch effects in Ratio-A, Ratio-G and BMC while also

overestimating batch effects in SVA (Fig. 2-3D). But it does overall, capture batch effects well (when present). The gPCA *P*-value, on the other hand, is usually significant only when gPCA deltas are close to 1. When the gPCA delta is low, however, the *P*-value range is usually insignificant. This is not useful, since it is only significant (and therefore, certain) when batch effects are confidently detected (in which case, we may simply rely on the delta), but never conveys confidence about the absence of batch effects. That is, we have not observed so far, a significant gPCA *P*-value associated with a gPCA delta close to 0. And so, it is sufficient to use large gPCA deltas alone as a proxy for strong batch effects. Similar results are also observed in RNA-seq data 2 and 3 and proteomics data (Appendix B: Figs. B1 to B3), and in the uniformity assumption (Appendix B: Figs. B4 to B7).

The representations in Fig. 2-3 (and Appendix B: Figs. B1 to B7) only convey the results for one simulation. To test for reproducibility and if the gPCA deltas are stable, we apply gPCA across 100 simulated RNA-seq datasets (this is modeled as 100 repetitions of the experiment). As shown in Fig. 2-4 (and Appendix B: Figs B8 to B14), the gPCA deltas are largely stable.

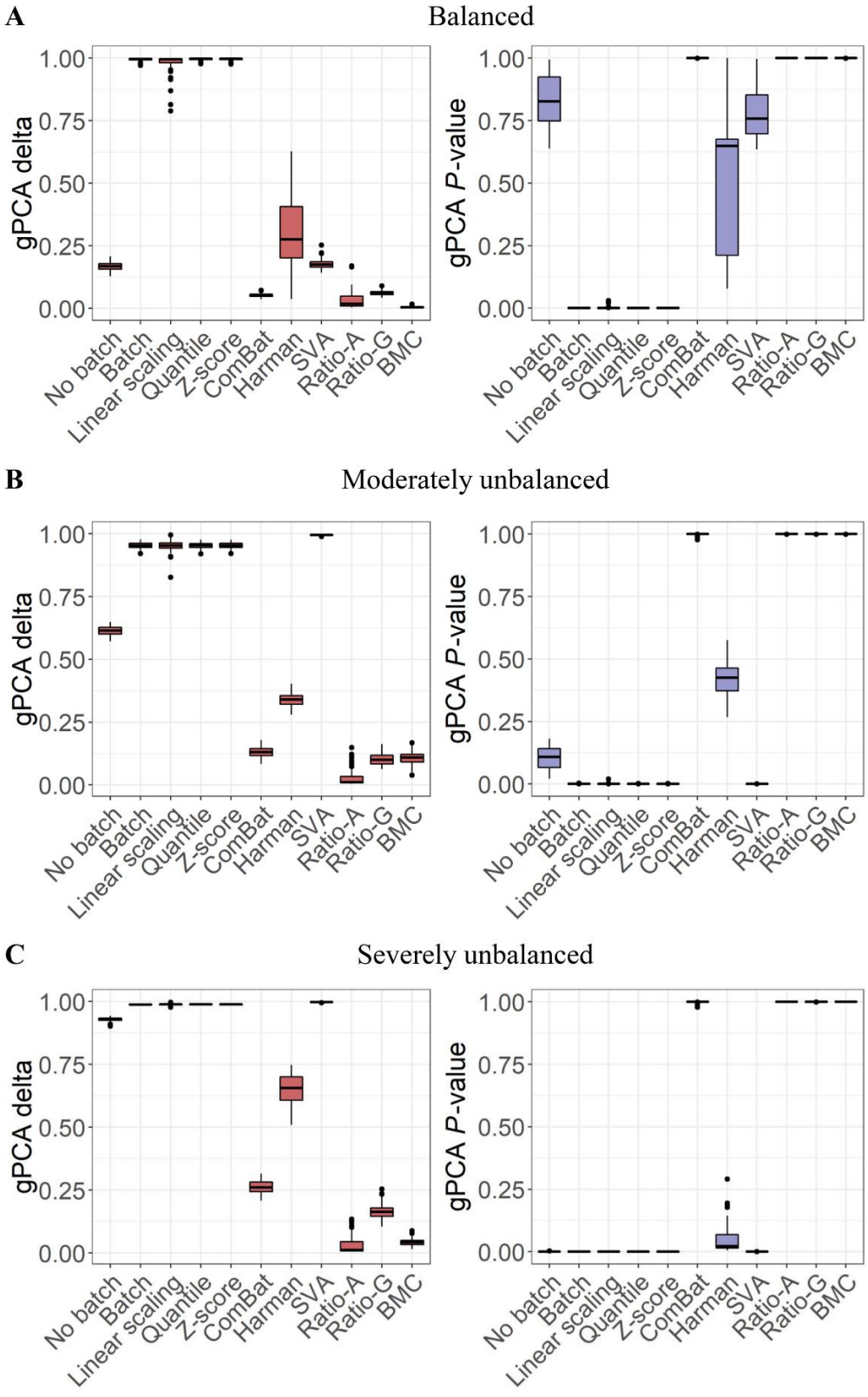


Fig. 2-4. gPCA delta and P-value distribution in batch-effect correction in RNA-seq data 1 (100 simulations). The batch-class effects are generated via the non-uniformity assumption.

As Fig. 2-4, the 100 simulations show that gPCA deltas become stably inflated in the moderate to unbalanced scenarios even before BEs are introduced. Under no circumstances do conventional normalizations (linear scaling, quantile and Z-score) ever reduce batch effects. The 100 simulations also stably suggest that ComBat, Ratio-A, Ratio-G and BMC methods work well while SVA is easily affected by class-batch unbalance. It also turns out that Harman's performance is quite variable, as evidenced by the rather large box-dimensions (the ends of the box denote the inter-quartile range). This suggests Harman is not likely to perform similarly across different datasets. The RNA-seq data 2 and 3 and proteomics data simulations suggest likewise (Appendix B: Figs. B8 to B10). And this is unlikely due to the batch-class simulation approach. Batch-class simulations using the uniformity assumption also show the same results (Appendix B: Figs. B11 and B14).

Use of the gPCA deltas facilitates rapid detection, quantitation and summarization of batch effects across many simulations. The interesting observation is that the results are largely stable, irrespective of platform (proteomics or transcriptomics) and batch-class simulation method. The only exception is Harman, which is sensitive to these Monte-Carlo type simulations. Although Harman's design is statistically sound, we suspect Harman's real-world performance may be unpredictable.

2.3.3 Seemingly good batch-effect correction is not a reliable indicator of good downstream statistical selection

Fig 2-3 (and Appendix B: Figs B1 to B7) reveals that visual representation and summary statistics for batch effects are not always congruent. Resolving batch effects is a difficult problem, since batch effects are complex, and may not be efficiently estimated and removed, and may produce false negatives and false positives[5, 12].

Since a set of true differential genes is known a priori, we may use the ability to correctly identify these as a practical performance gauge. In other words, suppose a BECA gives you low gPCA delta, but it fails at identifying the correct biological signals (genes), the BECA would still be worthless. It is possible a BECA may overcorrect due to strong overlap between class-batch variance, and this is a known problem with SVA[35].

There are two main approaches for evaluating statistical feature selection: The first is to rely on a single *P*-value cutoff (e.g., $P\text{-value} \leq 0.05$) but repeat the simulation many times, so that we may be able to approximate a good guess based on the average. This approach is easy to understand but does not capture "global performance", since a method that works poorly at a cutoff of 0.05 does not mean it will work poorly at other cutoffs such as 0.01 and 0.001. The second approach is to create a continuous graph showing the relationship between the True Positive Rate (sensitivity) and False Positive Rates (1-specificity). This is known as the receiver operating characteristic (ROC) curve. To help summarize the ROC, the area under the curve (AUC), bound between 0 and 1, is used instead. The problem with the ROC and AUC is that the differentiation may not be very big. Also, even with a good AUC, you will still need to identify a reasonable/suitable statistical cutoff in actual practice to identify differential genes for follow-up study.

Since both methods are imperfect, we will use both to check for corroboration. A BECA that is high for both F-score (the harmonic mean of precision and recall) and AUC, while also displaying low gPCA delta, robust against design imbalance, and consistent across our test datasets, is obviously ideal. And it will meet our requirement as a favored BECA (instead of just using a single performance metric).

When the batch-class is balanced (Figs. 2-5A and D), statistical feature selection results are aligned with expectations. Conventional normalization approaches produce lower F-scores and AUCs generally. All BECAs, despite differences in reported gPCA deltas, perform similarly in F-score and AUC evaluation. What this means is that even if the batch effects are not removed entirely, it may not adversely affect differential gene identification.

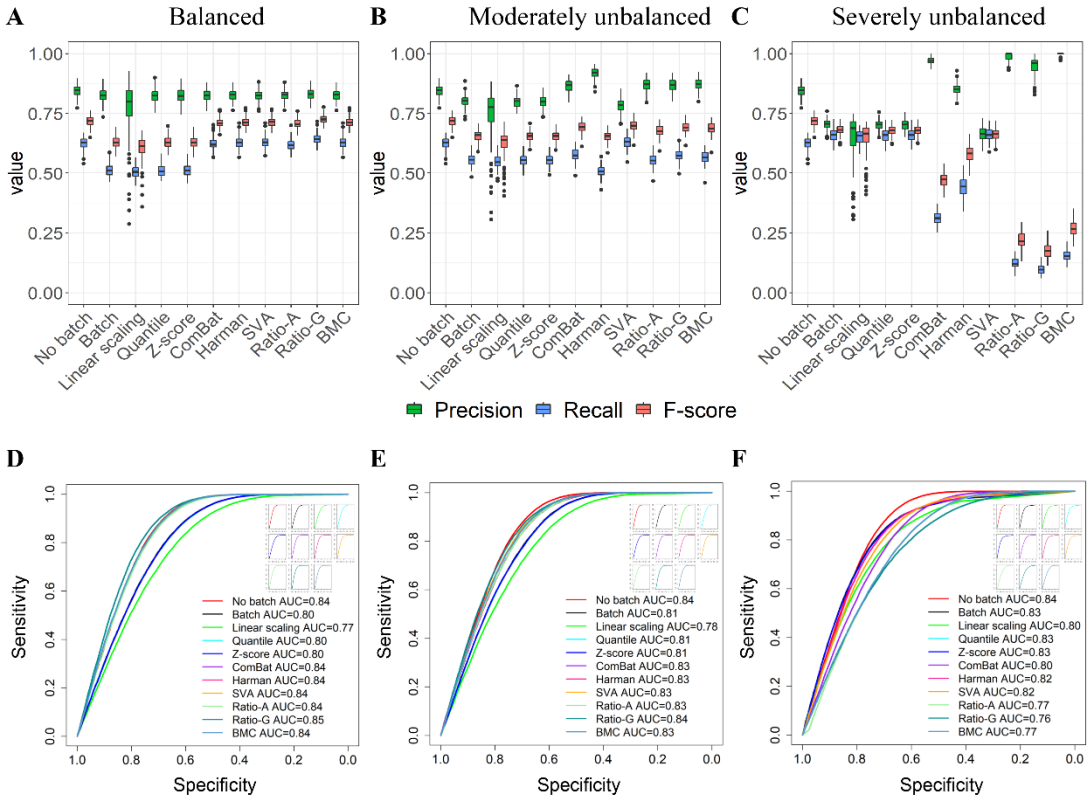


Fig. 2-5. Performance evaluation on precision/recall and ROC. **A-C**: The precision, recall and F-score distributions. **D-F**: The average ROC curves and the corresponding average AUC values. Results are based on RNA-seq data 1, non-uniformity assumption for batch-class, 100 simulations.

However, performance differentiation quickly emerges from moderate (Fig. 2-5B and E) to severely unbalanced (Fig. 2-5C and F) scenarios. In the moderate unbalanced scenario, while there is no clear winner in terms of F-score, Harman's precision moved up significantly (while its recall also dropped dramatically). SVA, which seems to have difficulty in removing batch effects when batch-class design is unbalanced, does not seem to have any issues with statistical feature selection, either F-score or AUC. It is performing well. The Ratio-A, Ratio-G and BMC methods are holding up well --- besides having low gPCAs, their performances are also good, and comparable with ComBat.

In the severely unbalanced scenario, SVA is surprisingly the winner for overall performance. This is due to a good balance between precision and recall, whereas other BECAs only manifest high precision, but very low recall (and therefore penalized due to the harmonic mean calculation). It should also be noted that although

the Ratio-A, Ratio-G and BMC methods are associated with incredibly low gPCA deltas, they have the worst performance in both F-score and AUC.

There is one final detail to note: Although conventional normalization does poorly in terms of batch-effect correction, it does little to hurt the quality of statistical feature selection. In fact, when dealing with severely unbalanced scenarios, conventional normalization actually produces better results than any BECA (with the possible exception of SVA).

These findings based on RNA-seq data 1 are also supported by our other test datasets (Appendix B: Figs. B15 to B17). The findings are also independent of the batch-class uniformity assumption (Appendix B: Figs. B18 to B21), although it should be noted that use of the uniformity batch-class assumption results in higher variability in the F-score distribution, this cannot be observed in the AUC because only the averaged curve is shown.

It appears that methods that deal with batch effects well (Ratio-A, Ratio-G, BMC, ComBat), do not necessarily perform well in statistical feature selection. In fact, this observation agrees well with Nygaard et al.'s observation[116] when study groups are unevenly distributed across batches, actual group differences may induce apparent batch differences, in which case batch adjustments may bias, usually deflate, group differences, and thus lead towards reduced sensitivity.

Although conventional normalization does not address batch effects issues, there seems to be little impact when it comes to statistical feature selection. In fact, when batch-class is severely imbalanced, it is preferable to not use any BECA at all. Also, as no BECA seems to be dominant in any one scenario so far, we are unable to conclude if a universally-best BECA exist. Finally, as we are only using log-transformed data so far (it is fine not to use any prior upstream normalization, as all simulated datasets are sampled from the same reference distribution anyway), it is useful to know if upstream normalization methods provide critical context in explaining differential performance amongst BECAs. If so, then it may help explain how certain benchmark experiments could be set up to skew the odds in favor of a certain BECA; but more importantly, help us understand which prior normalization method is synergistic with a given BECA.

2.3.4 Most BECAs are not particularly sensitive to upstream normalization procedures

Using Fig. 2-6, with cross-reference against Appendix B: Fig. B22 (RNA-seq data 2), Appendix B: Fig. B23 (RNA-seq data 3), Appendix B: Fig. B24 (proteomics) and simulations that do not use the batch-class non-uniformity assumptions (Appendix B: Figs. B25 to B28), we are unable to make many significant observations regarding optimal normalization-BECA pairs. In fact, simulations suggest that BECAs are not really affected by upstream normalization, which is a good thing since different studies may deploy different upstream conventional normalization methods to deal with technical variation and noise. We do notice that Ratio-A and Ratio-G consistently do not work well with the Z-score. And so, this pairing should probably be avoided.

Since upstream standardization does not seem to have much effect on performance, it may not be an important factor for ranking differences in BECA ranking studies. The performance differences observed between BECA may be due to the particular dataset studied, or the ranking is based on small differences (which we also observe here).

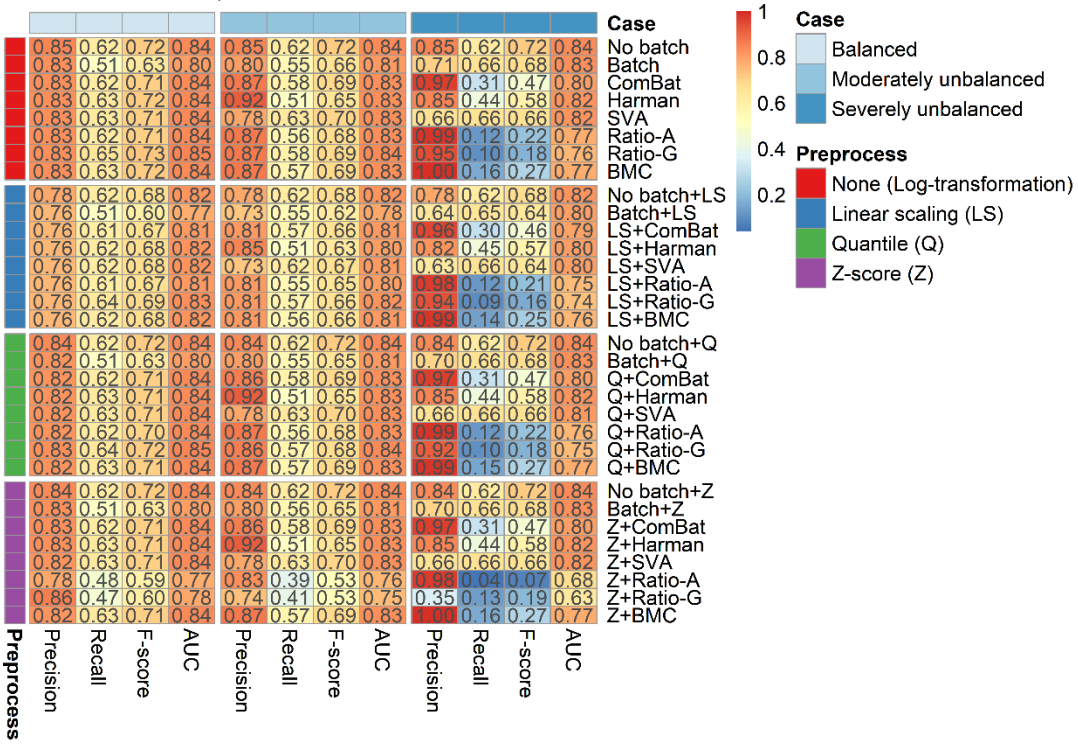


Fig. 2-6. The impact of prior normalization on BECAs. Results are based on RNA-seq data 1, non-uniformity assumption for batch-class, 100 simulations.

2.4 Discussion and conclusion

It is not entirely surprising that conventional normalizations perform poorly. After all, they do not explicitly correct for batch effects. We demonstrate this using quantile normalization (QN).

QN is a very widely used normalization technique. It is popular because it creates good standardization amongst samples. However, samples that look well-standardized do not mean that batch effects are corrected well. What QN is to calculate a mean for each ranked variable to get a distribution of means across the entire dataset. Then, each sample basically "inherits" this set of ranked means. Because each sample has the exact same set of ranked means (same set of numbers), they will have the exact same distribution. But that is not the same as having removed technical variation. Let's say we have sample 1 and 2 with two genes, A and B. Let's also introduce a technical bias to sample 2, and call it *C*. Let's assume a matrix (*M*), where the row and the column correspond to gene and sample, respectively.

$$M = \begin{bmatrix} 10 & 10C \\ 100 & 100C \end{bmatrix}$$

In QN, the first step is to rank the variables. Let's assume that A and B are already ranked, and B is 10x that of A. Next, we calculate an average for those variables occupying the same rank.

So for gene A, the average is $(10 + 10C)/2$. And for gene B, the average is $(100 + 100C)/2$. We then return these values back to the data matrix such that

$$M = \begin{bmatrix} (10 + 10C)/2 & (10 + 10C)/2 \\ (100 + 100C)/2 & (100 + 100C)/2 \end{bmatrix}$$

The technical effect *C* contributes directly to the ranked means after quantile normalization. If *C* is large, its contribution to the means will increase dramatically. It is part of the equation and ever present. In other words, QN does not remove batch effects.

Let us expand this discussion to cater for unbalanced design using QN again. Let's assume that all irrelevant genes have expression 0 and there is a single relevant gene G with true level $g < 0$ or $g > 0$ in class A and true level $g' < 0$ or $g' > 0$ in class B, and $g > 2 * g'$.

Suppose batch effects increase expression level 2x of all genes in the batch (note that irrelevant genes are still 0 levels). But let's say the batch design is unbalanced (75% class A and 25% class B are in the batch, while 25% A and 75% B not in the batch), though the classes are balanced (100 samples each).

Then the quantile normalized value of gene G in class A is g * (75% * 100) * 2 + g * (25% * 100) = g * (0.75 * 2 + 0.25) = 1.75 * g. And the quantile normalized value of gene G in class B is g' * 0.25 * 2 + g' * 0.75 = 1.25 * g'.

Given that $g > 2 * g'$, we can still clearly distinguish the normalized level of G in A (=1.75 * g > 3.5 * g') from that in B (= 1.25 * g'). That is, statistical feature selection can easily pick up G as differentially expressed gene (DEG).

However, as we have seen before, the batch effects are also not removed after normalization. Class A (B) in the batch has G's level as 2 * g (2 * g') and not in the batch as g (g').

Conventional normalization approaches such as Z-score, linear scaling, and quantile are non-effective at removing batch effects. And so, they should not be considered for such purposes.

If the experiment is balanced (i.e., class and batch factors are equally distributed), then it does not matter which BECA is used. They all perform similarly (with small differentials) in terms of precision and recall.

If the experiment is moderately unbalanced, and good precision is desired, Harman, ComBat, Ratio-A, Ratio-G or BMC is a good choice with acceptable recall (while Harman results in lower recall compared with others). If good recall is desired, then SVA is a good choice with acceptable precision.

If the experiment is severely unbalanced, the BECAs decline greatly in performance. However, if good precision is desired, ComBat is a good choice with acceptable recall. If good recall is desired, then SVA is a good choice with acceptable precision. Here, approaches such as Ratio-A, Ratio-G and BMC are too insensitive to be of any practical value.

When it comes to statistical feature selection stability, BECAs are generally insensitive to upstream normalization. However, note that Ratio-A is incompatible with Z-score.

A final recommendation, and only if possible, is to perform replication-type analyses, e.g., performing random samplings on the data-under-evaluation. Such

approaches help the analyst to evaluate the magnitude of a BE, and also to evaluate the stability of a BECA's performance. It is possible for replication-type analyses to help determine potential sources of confounding, which may or may not be due to batch effects. However, this point was not explicitly tested in our simulations as the batch factor was pre-determined.

BECAs are largely unaffected by upstream normalization procedures. While conventional normalization methods do not deal with batch effects at all, and they also do not affect downstream statistical feature selection. In less ideal situations where there are moderate levels of confounding, BECAs are robust and effective. Overall, there is no reason to believe a universal best BECA exist, this study suggests that BECAs are compromises in some way or another.

Chapter 3. More robust batch correction with ComBat in data with batch-class confounding issues

3.1 Introduction

High-throughput technologies are versatile tools used to systematically identify and/or quantitate small molecules in bio-samples (DNA, RNA, and protein, etc). Omics data are high-dimensional biological information derived from high-throughput technologies[2, 3]. Omics data are usually used in conjunction with statistical feature selection (SFS), with the aim of identifying a subset of phenotypically relevant features[117]. If performed effectively, such procedures have direct clinical implications in drug and biomarker development. However, effective SFS is reliant on data quality: if the majority of variance in data is associated with phenotype (also known as class effects), the majority of features identified by SFS should be phenotypically relevant[74, 81, 118]. Unfortunately, the presence of non-random technical variation can confound SFS. Collectively, these technical effects are referred to as batch effects[5, 12, 17].

Batch effects are a major challenge in current biological research[5, 15, 17], and are caused by systematic differences in processing time, laboratory, personnel, protocol, etc[15]. Although batch effects are minimizable via careful planning, it is often unavoidable as large samples need to be parallelized to save time or, adequate data volume is only achievable via collaboration between multiple laboratories. Various bioinformatics solutions have been developed to tackle batch effects[16], including Distance-Weighted Discrimination (DWD) based on Support Vector Machines (SVM)[13]; the Empirical Bayes (EB) method, ComBat[14]; Surrogate Variable Analysis (SVA)[60] and Remove Unwanted Variation (RUV)[61], both of which are based on Singular Value Decomposition (SVD). Batch mean-centering (BMC), which corrects batch effects via centering the variables within batches[64] and Ratio-based algorithms, include Ratio-A and Ratio-G[15] . Collectively, these methods are known as Batch Effect Correction Algorithms (BECAs)[5].

BECAs vary a lot: some require prior knowledge of the batch factors (the distribution of technical batches amongst samples), e.g. DWD[13], Ratio-A, Ratio-G[15], BMC[64] and ComBat[14]. Others do not and estimate this from the data itself, e.g. SVA[60] and RUV[61]. BECAs are imperfect solutions: careless use can create false effects, compromising data integrity[5, 12, 94]. Moreover, BECAs can also wrongly remove meaningful biological subpopulation information[5, 119]. It is therefore important to recognize that BECAs are not foolproof[61], and must be used appropriately[5, 12]. Although alternatives to BECAs exist, whether via Batch Effect Resistant Normalizations (BERN), e.g. Gene Fuzzy Scoring (GFS)[57] or via network-based analysis techniques [12], these methods are not perfect either. Both do not directly estimate and remove batch variation from the data, and also introduce further limitations, e.g. GFS only considers high-abundance genes/proteins while networks can only evaluate genes/proteins that overlap with it. Thus, BECAs remain highly relevant.

Although new BECAs emerge with routine regularity[62, 93], the EB approach, ComBat, remains popular due to its high reported performance[99]. However, ComBat do not always correct batch effects satisfactorily, especially when the batch-class design is unbalanced (where classes are not evenly distributed across batches), thus leading towards batch-class confounding[94, 116, 120-122]. Standard ComBat requires specification of batch labels but not class labels (i.e., ComBat will only remove the component of variance associated with the specified batch labels). Because ComBat is effectively blind to class information, it may not distinguish well the component of batch effects confounded with class. When data is presented to ComBat that includes class-associated variation (e.g. normal and disease phenotypes), the component of batch-associated variation confounded with class-associated variation is retained. Conversely, if data is presented to ComBat without class-associated variation, ComBat may do a more thorough job of removing all variation correlated with batch labels. Then we propose the class-specific ComBat (CS-ComBat) procedure, where data is split by class, presented to the ComBat software independently to remove batch effects, and then merged. To further address issues on batch-class confounding, we borrow from machine learning the use of Synthetic Minority Oversampling TEchnique (SMOTE)[123], a general strategy for dealing with class-imbalance problem, to explore synergies with BECAs given batch-class confounding issues.

CS-ComBat is benchmarked against other strategies for performing ComBat and with other three commonly used BECAs (Ratio-A, Ratio-G, and BMC). The three strategies for performing ComBat are “ComBat”, the quintessential ComBat approach that is performed on the whole dataset (i.e., only batch factors are specified); “ComBat (cov)”, which is similar to “ComBat”, but goes further to include class factors as covariate; “Class-specific”, which splits data first by class, on which ComBat is then performed independently on each split and then merging the batch-corrected splits. Ratio-A, Ratio-G, and BMC are also included. As they only require batch information for correction, they may present similar issues as ComBat when batch-class design is unbalanced. For all examined methods, we explore for potential synergies with SMOTE.

To determine the best strategy, we benchmark those BECAs before and after applying SMOTE on real and simulated gene expression data. We also consider both natural and simulated batch effects. In simulated data, we consider different degrees of confounding, class effect proportion (CEP) and batch effect proportion (BEP). We systematically measure BECAs performance using three different ways: partial redundancy analysis (pRDA) (for measuring batch effects and class effects), Jaccard coefficient (for evaluating inter-sampling similarity), and the student’s t-test followed by precision, recall and F-score (for evaluating SFS performance).

3.2 Materials and Methods

The analysis design is outlined in Fig. 3-1.

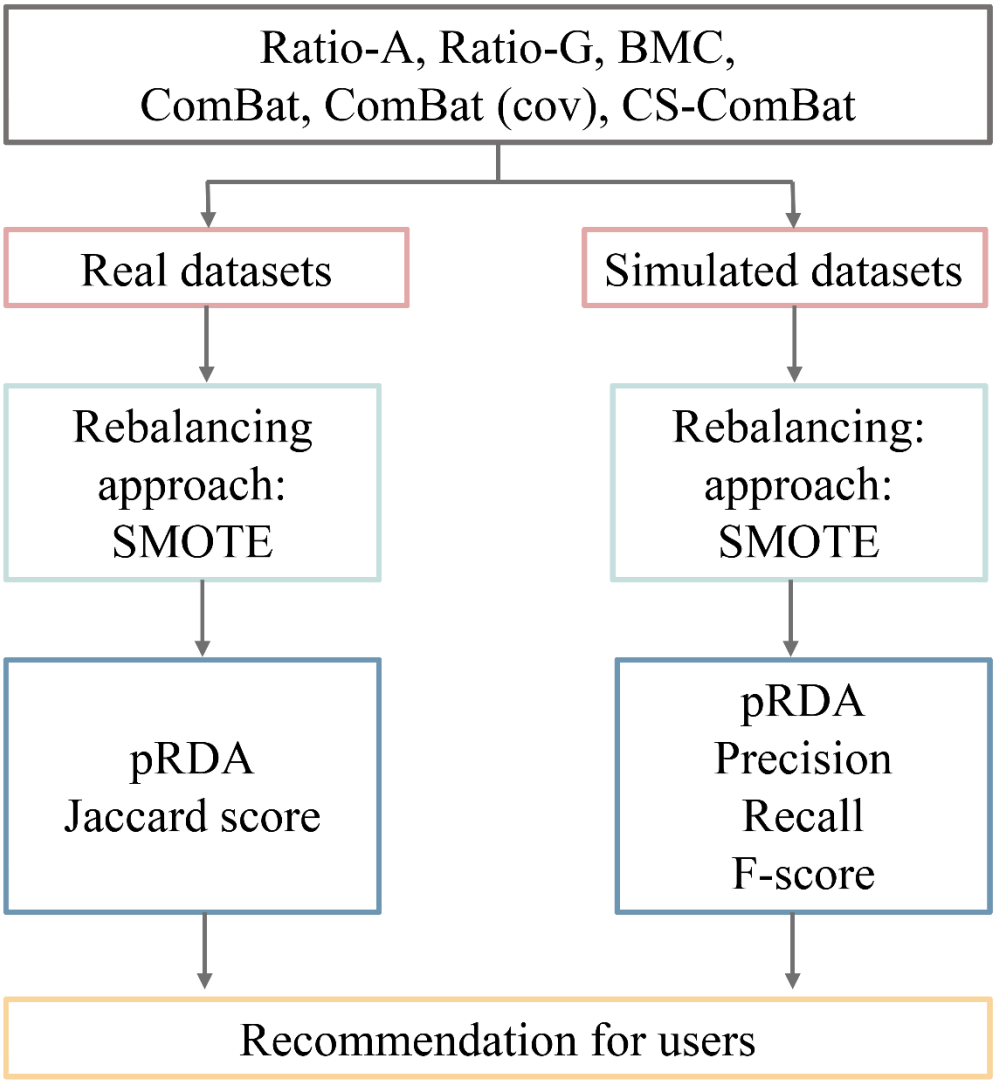


Fig. 3-1. Schematic overview of this study.

3.2.1 Real data

To generate the data with real batch effects, we used previously published datasets: GSE22544[124] and GSE8977[125] available from the Gene Expression Omnibus (GEO)[24]. Data information are listed in Table 3-1. They are studies from two different labs, then each data can consider as a single batch, real batch effects can be created through merged them together. Since classes are unevenly distributed in batches, the batch-class design is unbalanced, there is confounding exist in the merged data (R^2 = 0.48). R^2 is often used to explain the fit of linear regression, where it is

used in here to indicate the degree of confounding between class and batch effects. When $R^2 = 0.48$, which means there is 48% confounding between class and batch factors.

The pre-processing step involved background correction and data annotation. Background correction of the raw values was handled with the R package "oligo"[126]. Data annotation was conducted with NetworkAnalyst web-based tool[127]. In particular, probes were mapped to the entrez ID, when multiple probes mapping to one entrez ID, the mean gene expression value of these probes was used.

Table 3-1. The data information. These two data were combined to generate real batch effects. SMOTE used to make the data become balanced.

Number	Dataset	Platform	Country	Types of sample	Control *vs* case (Before SMOTE)	Control *vs* case (After SMOTE)	Reference
1	GSE22544	Affymetrix Human Genome U133 Plus 2.0 Array	USA	ductal of the breast	4 *vs* 16	16 *vs* 16	[124]
2	GSE8977	Affymetrix Human Genome U133 Plus 2.0 Array	USA	ductal of the breast	15 *vs* 7	15 *vs* 15	[125]

3.2.2 Simulated data

We simulate data modeled from the parameters of real data: An RNA-Seq dataset with three pairs of case and control samples[106].

The R package "polyester" was used to estimate the dispersion parameters from the real RNA-seq data, and then fitted into a negative binomial distribution[110]. We

simulated 100 simulations. In each simulation, we simulated the gene matrix with 1000 genes and 80 samples without class effects and batch effects.

In this study, three scenarios of confounding are simulated: (1) batch-class design is balanced ($R^2 = 0$); (2) batch-class design is moderately unbalanced ($R^2 = 0.4$); (3) batch-class design is severely unbalanced ($R^2 = 0.8$).

We simulated the class and batch effects based on the approach of Leek[76]. For each simulation, 80 samples are assigned to two classes (40 samples in each class, respectively). For the genes, the class factors (class A and class B) were drawn from a normal distribution that had mean of 0 and standard deviation of 1 and simulated on 20%, 50%, and 80% of total genes by using the R package of "polyester". Since class factors are simulated, then the differential genes/proteins are known a priori.

After incorporating class effects, the 80 samples are split into 2 technical batches. The batch allocation is according to the batch-class design (balanced, moderately unbalanced or severely unbalanced), batch factors were also drawn from a normal distribution with mean 0 and standard deviation 1 and simulated on 20%, 50%, and 80% of total genes by using R package of "polyester". Because batch effects are simulated only on a subset of genes, this is referred to as a non-uniform approach.

The simulation design of different degrees of confounding, class-effect proportion (CEP) and batch-effect proportion (BEP) was presented in Fig.3-2.

A Before SMOTE

	Balanced		Moderately unbalanced		Severely unbalanced	
	Class 1	Class 2	Class 1	Class 2	Class 1	Class 2
Batch 1	20	20	28	12	36	4
Batch 2	20	20	12	28	4	36

B After SMOTE

	Class 1	Class 2	Class 1	Class 2	Class 1	Class 2
Batch 1	20	20	28	28	36	36
Batch 2	20	20	28	28	36	36

C

CEP (%)	BEP (%)	Genes affected only by class effects (%)	Genes affected only by batch effects (%)	Genes affected by class and batch effects (%)
20	20	10	10	10
	50	10	40	10
	80	10	70	10
50	20	40	10	10
	50	10	10	40
	80	10	40	40
80	20	70	10	10
	50	40	10	40
	80	10	10	70

Fig. 3-2. Simulation design. Simulated data with different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced") before SMOTE (**A**) and after SMOTE (**B**), class effects proportion (CEP) and batch effects proportion (BEP) (**C**).

3.2.3 Data pre-processing

3.2.3.1 Log-transformation

Before batch effects correction, data are applied with log-transformation for reducing scale and making the data with better distribution symmetry[76].

3.2.3.2. Rebalancing approach: Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE[123] is an oversampling technique that uses the information available in the data to generate synthetic samples from a minority class. SMOTE was deployed to handle class imbalanced when batch-class confounding, SMOTE implemented in R "UBL" package (https://cran.r-project.org/web/packages/UBL/index.html).

In both tested real and simulated data, when batch-class design is unbalanced, SMOTE was applied to generate balanced data. Then the balanced data is used for

batch effects correction with different BECAs. After that, those oversampling samples generated by SMOTE were removed (the aim is to keep the degree of freedom without changing, then making a fair comparison in the statistical feature selection), then performing batch effects detection and the following performance evaluation.

3.2.4 Batch effects correction algorithms (BECAs)

3.2.4.1 ComBat (Software)

ComBat is a powerful method for correcting batch effects, especially on small sample size. It is based on the Empirical Bayes framework using estimations for the location (mean) and /or scale (variance) parameters for each gene [14].

Let Y_{ijg} indicate the gene expression value of gene g in sample j in batch i, then it is assumed that the Y_{ijg} can be expressed as the defined location and scale model equation [14]:

$$Y_{\text{ijg}} = \alpha_g + X\beta_g + \gamma_{ig} + \delta_{ig}\varepsilon_{ijg}$$

Where α_g indicates the overall gene expression, X represents the design matrix for sample conditions, β_g is the vector of regression coefficients corresponding to X, γ_{ig} and δ_{ig} are the additive and multiplicative batch effects of batch i for gene g, respectively. The noise terms are indicated by ε_{ijg}, and it is assumed to follow a normal distribution with mean zero and variance σ_{g}^2 [14].

The standardized data of Z_{ijg} is assumed to be normally distributed, $Z_{\text{ijg}} \sim N(\gamma_{ig}, \delta_{ig}^2)$, where γ_{ig} and δ_{ig}^2 are estimates of the batch effects parameters with the prior distributions of Normal and Inverse Gamma, respectively [14]. Given by

$$Z_{\text{ijg}} = \frac{Y_{ijg} - \hat{\alpha}_{\text{g}} - X\hat{\beta}_g}{\hat{\sigma}_g}$$

After applying the EB to estimate the batch effects parameters, then the batch effects adjusted data γ_{ijg}^* is calculated by

$$\gamma_{\text{ijg}}^* = \frac{\hat{\sigma}_g}{\hat{\delta}_{ig}^*}\left(Z_{ijg} - \hat{\gamma}_{ig}^*\right) + \hat{\alpha}_g + X\hat{\beta}_g$$

The ComBat R package is available from the BioConductor package “sva” [112].

3.2.4.2 ComBat, ComBat (cov) and CS-ComBat (Running mode)

The ComBat referred to here, is the generic mode of ComBat use, not the software itself. In this mode, the entire dataset with known batch factors (e.g., the time of test) are presented to the ComBat software for batch cleaning.

For the "ComBat (cov)" mentioned in here is to specify both batch and class factors to ComBat, and class factors are incorporating as a covariate.

The class-specific ComBat mode, or CS-ComBat, is a modification to standard procedure. It involves splitting data first by class (retaining the batch factors), cleaning each split of the data by ComBat software individually, and re-merging to produce the final batch-cleaned dataset.

3.2.4.3 Ratio-based methods (Ratio-A and Ratio-G)

In ratio-based methods, there are two types that existed, one is ratio-based method by using arithmetic mean as reference (Ratio-A), the other is ratio-based method by using geometric mean as reference (Ratio-G)[15]. The Ratio-A and Ratio-G are available from the "bapred" R package[80].

3.2.4.4 Batch mean-centering (BMC)

In BMC, it converts the data by doing batch-wise subtracting of the variables by their arithmetic means, therefore, the mean of each gene becomes zero[64]. The BMC is available from the R "bapred" package[80].

3.2.5 Batch effects detection with partial redundancy analysis (pRDA)

The pRDA can be used to calculate the variance associated with the batch effects and class effect. It first fits multivariate linear regression to data and the controlled variables (e.g., batch factors or class factors), then perform PCA analysis to summary the variation associated with class factors or batch factors[92, 128]. The pRDA available from the "vegan" R package[129].

3.2.6 Performance evaluation on real data

In the merged real data, where the set of DEGs are unknown in a prior, the performance was tested by using alternative methods[58]. In particular, we resample at different resampling sizes (4, 6, and 8) for each set of classes (normal vs cancer) given each approach. Random subsampling is conducted 1000 times to generate a

binary matrix, where rows and columns indicate samplings and genes, respectively. A value of 1 denotes statistical significance (based on the standard t-test with *P*-value cutoff at 0.05) and 0 otherwise.

For inter-sampling similarity, we used the Jaccard coefficient to compare the significant genes selected in each resampling. Let A1 and A2 be the DEGs selected by independently applying feature selection method (based on the standard t-test with *P*-value cutoff at 0.05) on the two sets of classes (normal vs cancer) given two resamplings.

Then, we may use the Jaccard coefficient to measure the inter-sampling similarity:

$$Jaccard\ coefficient = \frac{|A1 \cap A2|}{|A1 \cup A2|}$$

Jaccard coefficient range from 0 to 1, the higher the value indicates the better inter-sampling similarity.

For feature-selection stability, we summed and normalized each column according to the number of resampling, then a value near to 1 indicates high stability. The distribution of genes stabilities offers general characteristics of the stability of the test method.

3.2.7 Performance evaluation on simulated data

3.2.7.1 True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN)

Since in the simulated data, we know the set of differential variables in a priori, then we can test the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) to evaluate BECAs performance before and after SMOTE.

3.2.7.2 Statistical feature selection analysis

To evaluate BECA-associated impact on data integrity (i.e., the creation of false effects leading to false positives or false negatives due to the use of BECAs on data), we used feature selection (based on the standard t-test with *P*-value cutoff at 0.05) followed by precision, recall and F-score measurement. In this study, we did not apply multiple test corrections since they will artificially improve precision while penalizing recall, which leading the evaluation becomes unfair.

Precision and recall may be expressed by the formulae below:

$$Precision = \frac{TP}{TP + FP}; Recall = \frac{TP}{TP + FN}$$

Where TP, FP and FN are the true positives, false positives and false negatives respectively. The F-score (F_s) is the harmonic mean of the precision and recall used for estimating the overall performance:

$$F_s = 2 * \frac{Precision * Recall}{Precision + Recall}$$

3.3 Results

We evaluated three ComBat-strategies and another three commonly used BECAs based on real and simulated gene expression data treated with or without SMOTE. For real data, natural batch effects are generated by combining samples from two data together (where each data represents a single batch, Table 3-1). Since the set of true differential expression genes (DEGs) are not known a priori, performance for inter-sampling similarity may be evaluated via the Jaccard coefficient. For simulated data, batch effect simulation approaches are described in **Materials and Methods**. Since the true DEGs are known a priori, performance for good signal recovery may be evaluated on precision, recall and F-score (see **Materials and Methods**). An overview of this study is provided in Fig. 3-1.

3.3.1 On real gene expression data, CS-ComBat excels at class effect preservation

Natural batch effects are inducible by merging independently acquired data. Barplots are used for showing the variance explained by batch or class effects calculated with pRDA for each BECA, such that if a BECA corrected batch effects well, the variance explained by class should be larger than batch. As shown in Fig. 3-3A, where batch-class design is unbalanced (before SMOTE) presenting a challenging scenario for BECAs, CS-ComBat performs the best in terms of preserving class effects compared with other BECAs: CS-ComBat with a variance value explained by class of 0.29, while other BECAs (Ratio-A, Ratio-G, BMC,

ComBat and ComBat (cov)) had a variance explained by class between 0.05 and 0.08. Unfortunately, CS-ComBat is less powerful than other BECAs in removing batch effects. However, after SMOTE (Fig. 3-3B), CS-ComBat not only could remove batch effects as powerful as other BECAs (with the variance of batch around 0.03), but remains the top method for preserving class effects (with at least 1.6 fold variance *vs* other BECAs), amongst all tested BECAs.

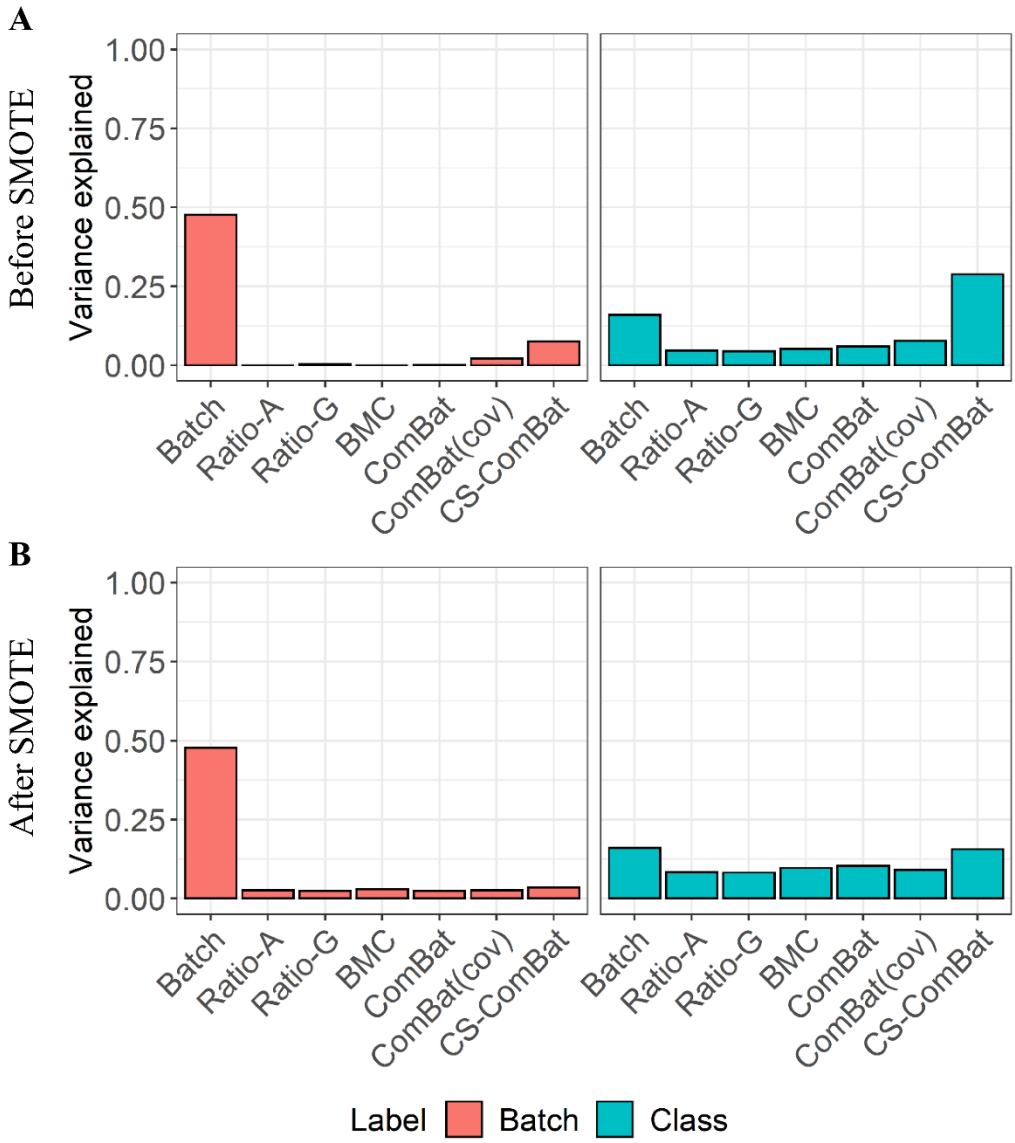


Fig. 3-3. The variance explained by batch or class was detected with pRDA before SMOTE (**A**) and after SMOTE (**B**). The x-axis shows each method. The y-axis is the variance explained by batch (red) or class (blue).

3.3.2 On real gene expression data, evaluation based on Jaccard coefficient suggests that CS-ComBat produces highly reproducible results

Next, to understand the impact of batch effect correction on the data, we evaluate inter-sampling similarities via the Jaccard coefficient[58]. We sample sizes of 4, 6 and 8 to mimic small, medium and large sample size scenarios. Prior to SMOTE,

CS-ComBat performs the best on inter-sampling similarity across all sampling scenarios (CS-ComBat: more than 3 fold Jaccard coefficient *vs* other BECAs) (Fig. 3-4A). In addition, CS-ComBat selects highly similar feature sets even from very small samplings (with a sampling size of 4) from the original data compared to the other BECAs, suggesting suitability for small sample-size analysis. This is also supported by feature-selection stability (Appendix C: Fig. C1). After applying SMOTE to rebalance samples, although CS-ComBat's Jaccard coefficient and feature-selection stability declines, it remains the best over other BECAs (Fig. 3-4B and Appendix C: Fig. C1). Interestingly, after SMOTE, the Jaccard coefficient of Ratio-A, Ratio-G, BMC and ComBat increased, and their performance approximates ComBat (cov).

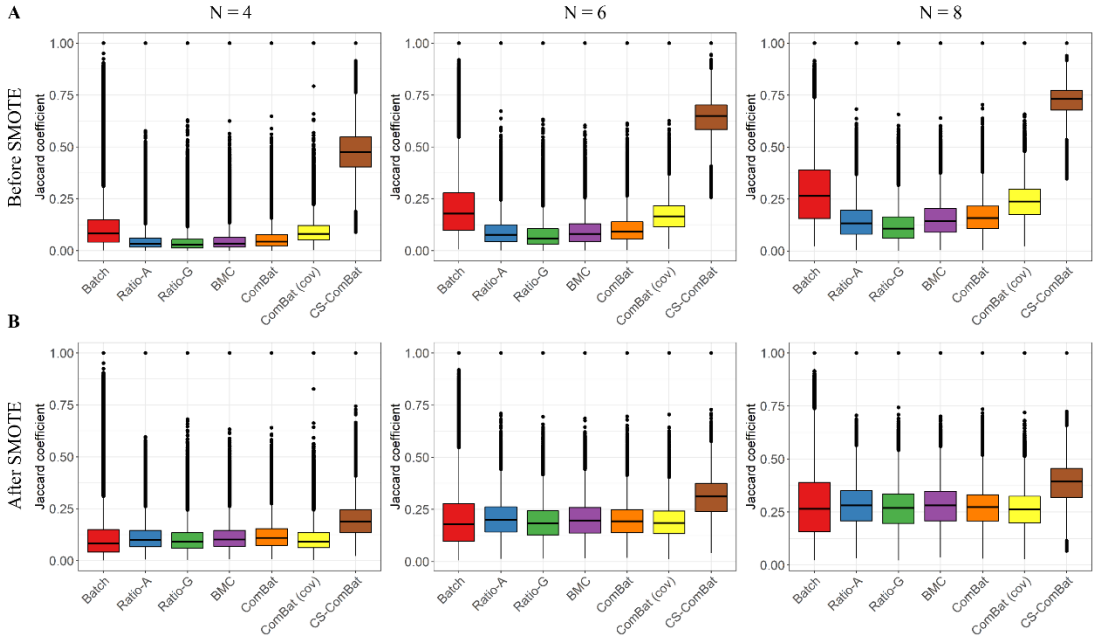


Fig. 3-4. Inter-sampling similarity before SMOTE (**A**) and after SMOTE (**B**). The x-axis shows each method. The y-axis is the Jaccard coefficient. N indicates the sampling sizes of 4, 6, and 8. Results based on 1000 times of resampling.

3.3.3 On Simulated gene expression data, evaluation based on pRDA also suggests that CS-ComBat excelling at class effects preservation

On real data, the set of true DEGs not known *a priori*. Thus we further investigate the performance of CS-ComBat and other BECAs on simulated data. Performance was investigated under combinations of different degrees of

confounding, class effects proportion (CEP) and batch effects proportion (BEP), with and without a rebalancing approach (SMOTE) (Fig. 3-2).

Firstly, the performance of CS-ComBat and other BECAs in terms of batch effect correction (Fig. 3-5) and class effect preservation (Fig. 3-6), with or without SMOTE were studied using pRDA. For batch effect correction before SMOTE (Fig. 3-5A)**,** all tested BECAs including CS-ComBat remove batch effects well when batch-class design is balanced (with a mean variance value explained by batch close to 0.00). However, performance differentials emerge when batch-class design become unbalanced (moderately unbalanced and severely unbalanced). In this case, CS-ComBat was less powerful in removing batch effects than other BECAs. Besides, note that when batch-class design is unbalanced, although in the no batch case where only class effects are simulated but not batch effects, the pRDA shows both class and batch effects exist due to confounding between class and batch factors. After SMOTE (Fig. 3-5B), the ability of CS-ComBat in removing batch effects was enhanced compared with analyses before SMOTE, with a smaller mean variance explained by batch, especially when BEP is larger than CEP. For example, the mean variance value explained by batch of CS-ComBat decreased from 0.29 (before SMOTE) to 0.15 (after SMOTE) when batch-class is severely unbalanced under CEP is 20% and BEP is 80%. While the ability to remove batch effects by other BECAs (ComBat, Ratio-A, Ratio-G and BMC) decreased when applying SMOTE in the unbalanced data, with a larger mean variance explained by batch compared to that in the dataset before SMOTE.

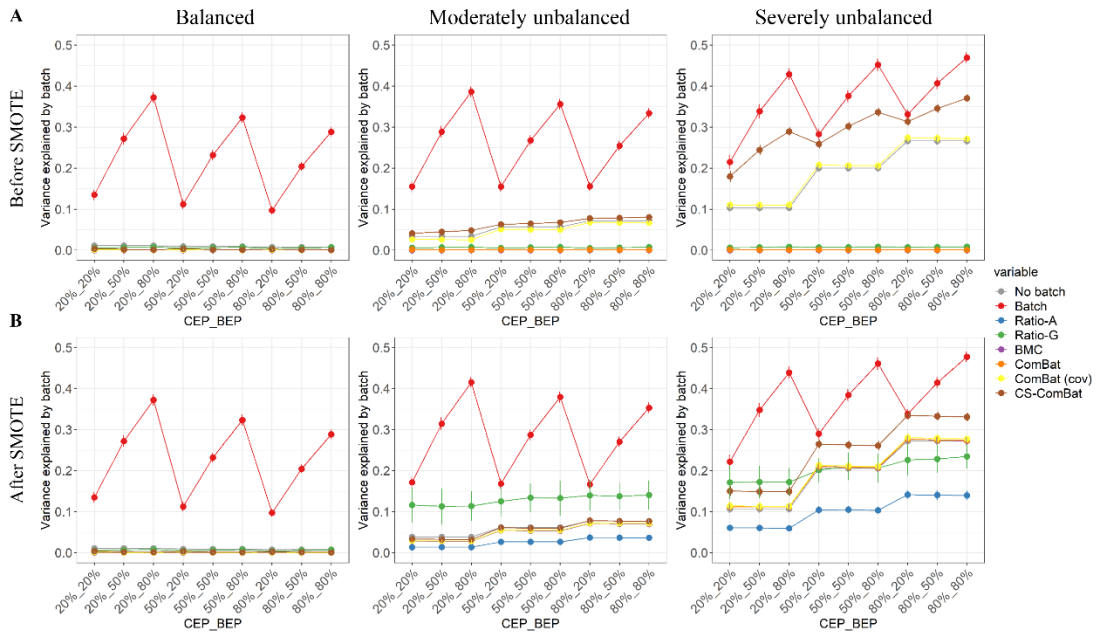


Fig. 3-5. Performance evaluation based on variance explained by batch calculated with pRDA of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion and batch effects proportion ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

In terms of class effects (Fig. 3-6), CS-ComBat performed the best in preserving class effects among tested BECAs, especially when batch-class is severely unbalanced. Before SMOTE (Fig. 3-6A), CS-ComBat showed the largest mean variance value explained by class (0.56), followed by ComBat (cov) with a mean variance explained by class of 0.42, and other BECAs had a mean variance value explained by class between 0.05 and 0.08 when batch-class is severely unbalanced under CEP is 80% and BEP is 80%. After SMOTE (Fig. 3-6B), although the ability of CS-ComBat in preserving class effects decreased compared with analyses before SMOTE (and when BEP is larger than CEP under batch class design is unbalanced), CS-ComBat still performed the best in preserving class effects among tested BECAs. On the other hand, the ability to preserve class effects of other BECAs (ComBat, Ratio-A, Ratio-G and BMC) were increased when compared with those in dataset before SMOTE. pRDA indicated that those BECAs (ComBat, Ratio-A, Ratio-G and BMC) which do not consider class factor tends to remove both batch effects and class effects when batch-class design is severely unbalanced, while CS-ComBat and

ComBat (cov) tend to preserve as much as the class effects although at the cost of retaining some batch effects.

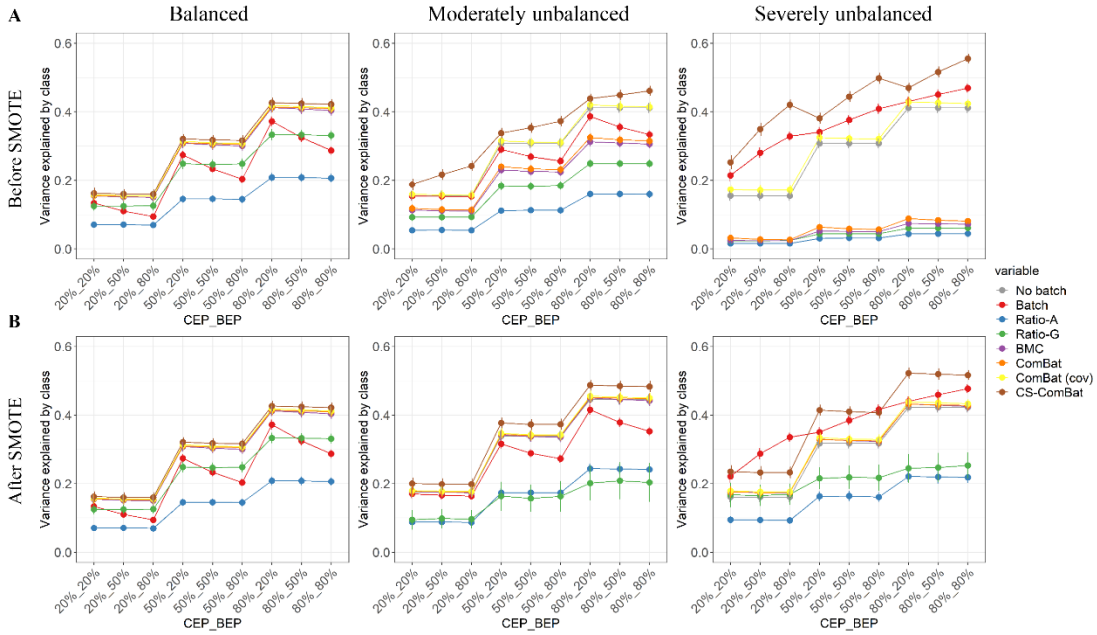


Fig. 3-6. Performance evaluation based on variance explained by class calculated with pRDA of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion and batch effects proportion ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

3.3.4 On Simulated gene expression data, evaluation based on statistical feature selection suggests that CS-ComBat excels at sensitivity (recall)

Next, we further investigated the performance of all tested BECAs in terms of statistical feature selection (precision, recall and F-score) on simulated data given different degrees of confounding (between the class and batch factors), class-effect proportion (CEP) and batch-effect proportion (BEP), and finally, with and without a rebalancing approach (SMOTE).

In terms of precision (Fig. 3-7), where batch-class design is balanced, all tested BECAs had good performance on simulated data before and after SMOTE. However, when batch-class design is unbalanced (moderately and severely unbalanced), the precision values of CS-ComBat were lower than other BECAs before SMOTE (Fig.

3-7A). In addition, CS-ComBat and ComBat (cov) is relatively better when CEP is much larger than BEP, than when BEP is much larger than CEP, suggesting that CEP and BEP are critical factors affecting precision. After SMOTE (Fig. 3-7B), when batch-class design is unbalanced, the precision of CS-ComBat increased, especially under the case of BEP is larger than CEP, while other BECAs (ComBat, Ratio-A, Ratio-G and BMC) decreased. ComBat (cov) seems unaffected by SMOTE: it performs similar before and after SMOTE. An interesting observation is that after SMOTE, all BECAs perform similarly in terms of precision. For example, the tested BECAs had a mean precision between 0.89 and 0.94 when batch-class is severely unbalanced under CEP is 80% and BEP is 80%.

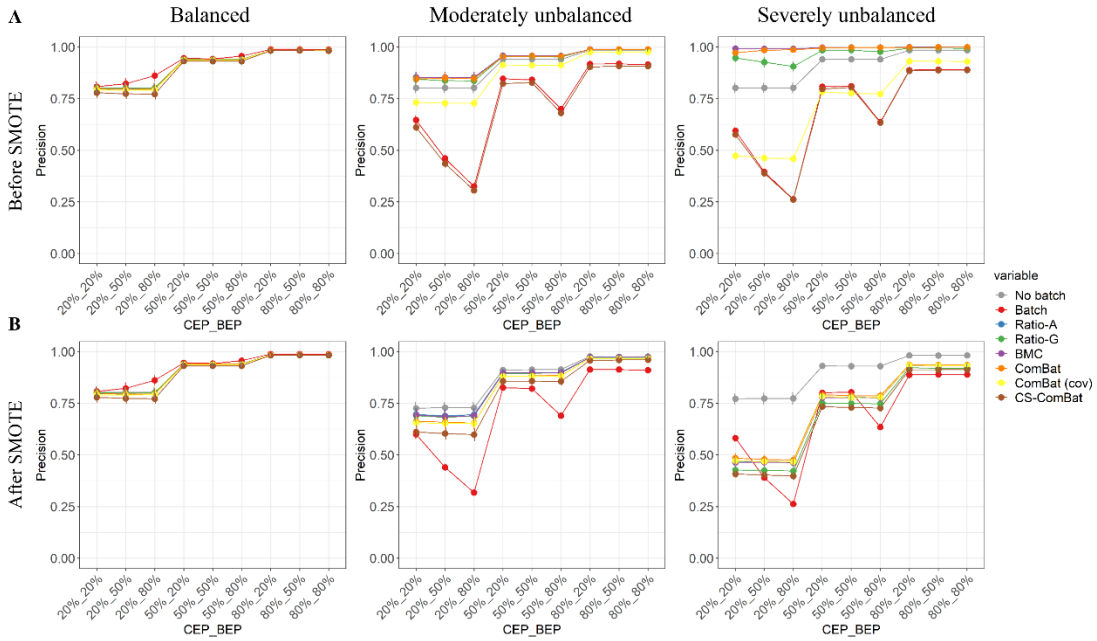


Fig. 3-7. Performance evaluation based on the precision of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion and batch effects proportion ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

In terms of recall (Fig. 3-8), in balanced case, all tested BECAs including CS-ComBat had good performance on simulated data before and after SMOTE. However, when batch-class design is unbalanced, especially in the severely unbalanced case, CS-ComBat was the most powerful method among all tested BECAs on simulated data before SMOTE (Fig. 3-8A). For example, CS-ComBat had

a mean recall of 0.84, and the recall of ComBat (cov) was 0.81, while the corresponding number for other BECAs (Ratio-A, Ratio-G, BMC, and ComBat) was just around 0.50 when batch-class is severely unbalanced, indicating the especially unstable properties (in terms of recall) of Ratio-A, Ratio-G, BMC, and ComBat against confounding effects. After SMOTE (Fig. 3-8B), all tested BECAs give similar recall. For example, they reported a mean recall between 0.80 and 0.85 when batch-class is severely unbalanced. Generally, CS-ComBat could produce a reasonable recall on simulated data under different degrees of confounding, CEP and BEP, with and without a rebalancing approach (SMOTE).

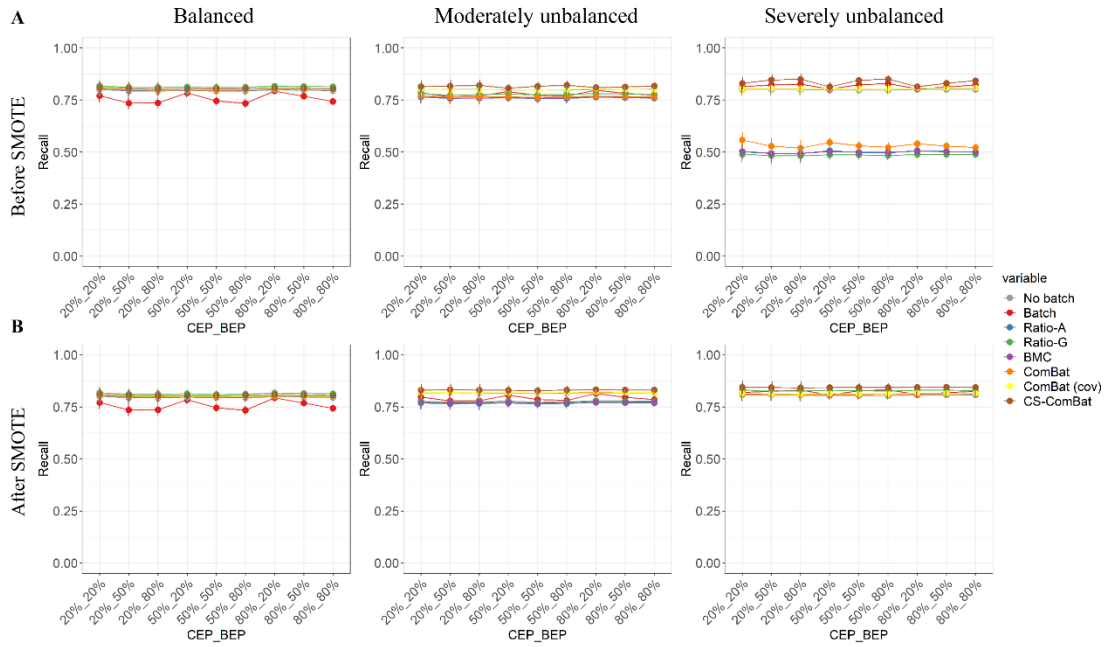


Fig. 3-8. Performance evaluation based on recall of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion and batch effects proportion ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

In terms of F-score (Fig. 3-9), under the balanced case, all tested BECAs including CS-ComBat performed well on simulated data before and after SMOTE. CS-ComBat and ComBat (cov) were also more powerful than other methods when batch-class is severely unbalanced and CEP is medium (50%) or high (80%) (Fig. 3-9A). For example, CS-ComBat and ComBat (cov) with the same mean F-score of 0.87, compared to other BECAs that had a mean recall between 0.66 and 0.69. When

CEP is low (20%) and BEP is high (80%) and the batch-class design is unbalanced, CS-ComBat and ComBat (cov) reports a lower F-score compared to other BECAs, for example, CS-ComBat and ComBat (cov) with mean F-score of 0.40 and 0.59, respectively, compared to the other BECAs that had a mean F-score between 0.63 and 0.68 when batch-class is severely unbalanced under CEP is 20% and BEP is 80%. After SMOTE (Fig. 3-9B), when batch-class design is unbalanced, the F-score of CS-ComBat increased, especially in the scenario where BEP is larger than CEP, while other BECAs (ComBat, Ratio-A, Ratio-G and BMC) with some decrease. ComBat (cov) seems not affected by SMOTE, it performs similar before and after SMOTE. Besides, after SMOTE, all the BECAs perform similar in terms of F-score. For example, the tested BECAs had a mean F-score between 0.86 and 0.89 when batch-class is severely unbalanced under CEP is 80% and BEP is 80%.

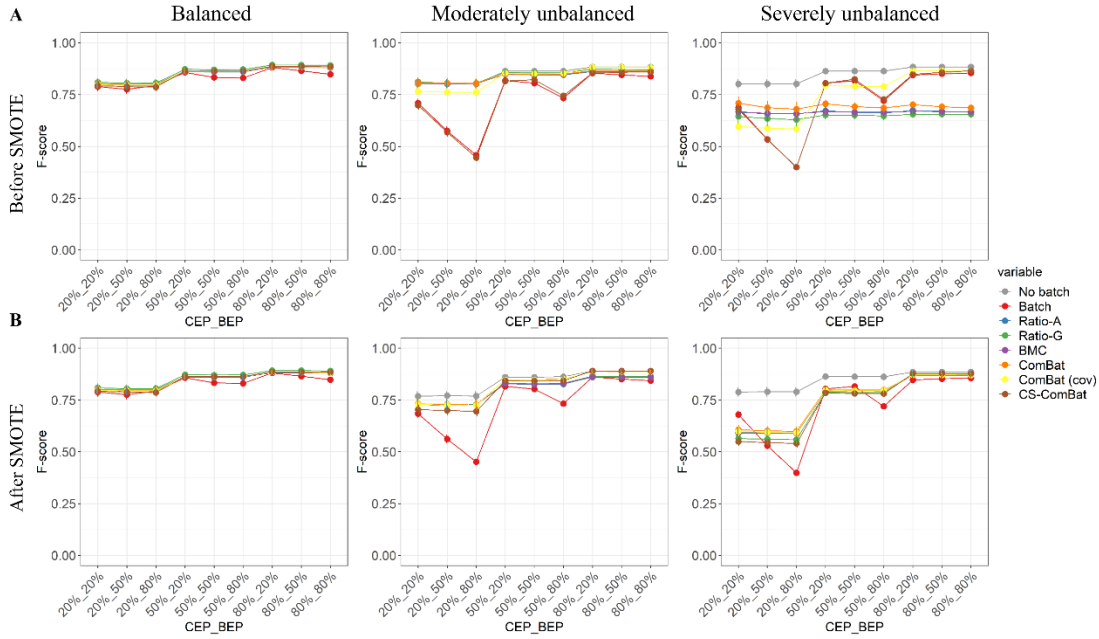


Fig. 3-9. Performance evaluation based on F-score of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion and batch effects proportion ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

Since in the simulated data, we know the true DEGs *a priori*, then the ability to correctly identify these genes may be used as a practical performance indicator. When batch-class design is balanced, all the tested BECAs with good performance in terms

of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), while performance differences appear when batch-class design is unbalanced, especially in the severely unbalanced case (Appendix C: Figs. B2-B5). Before SMOTE, when batch-class design is severely unbalanced, CS-ComBat and ComBat (cov) were more powerful than other BECAs when CEP is high (80%). ComBat (cov) and CS-Combat, which seems to have difficulty in removing batch effects in the severely unbalanced case, but do not have issue in correctly identify those genes as TP, TN, FP, FN, and they are performing well. While Ratio-A, Ratio-G, BMC and ComBat results in lower TP and FP but higher FN. In addition, there is a limitation for CS-ComBat and ComBat (cov), they tend to result in higher FP compared with other BECAs when CEP is low (20%) while BEP is high (80%). However, after SMOTE, the FP of CS-ComBat was decreased compared to before SMOTE, especially under the case when CEP is low (20%) while BEP is high (80%). Also, after SMOTE, in terms of TP, TN, FP and FN, the performance of different BECAs becomes stable and similar from balanced to the severely unbalanced case.

Taken together, CS-ComBat and ComBat (cov) perform as well as other BECAs when batch-class design is balanced, but also perform better than other BECAs when batch-class design is unbalanced under high CEP. However, there are limitations of ComBat (cov) and CS-ComBat, when CEP is low (20%) while BEP is high (80%), CS-ComBat tends to make some FP, leading to lower precision and F-score compared with other BECAs. CS-ComBat has not been suggested used alone in this case. Besides, SMOTE can make the performance of different BECAs becomes quite stable and similar, and they are so similar that it no longer matters what kind of methods is used.

3.4 Discussion and conclusion

Separating batch effects from real changes due to phenotype, to avoid the risk of removing meaningful information is a challenge when the batch-class design is unbalanced, and this problem is even more serious when the data is severely unbalanced. Under this scenario, applying ComBat on the whole dataset with only specific batch factors, is not effective, tends to remove a lot of class information (a similar problem for Ratio-A, Ratio-G and BMC as those BECAs only consider the

batch factors). They are not recommended to direct use. Fortunately, using a CS-ComBat or ComBat (cov) instead of those BECAs can have a significant positive impact on performances when batch-class design is unbalanced, especially in the severely unbalanced case. In addition, SMOTE is a useful strategy in coordinated with BECAs to rescue their performance in the severely unbalanced case under medium CEP to high CEP. SMOTE can make the performance of different BECAs becomes quite stable and similar, and they are so similar that it no longer matters which BECA is used.

The performance of three strategies of doing ComBat (ComBat, ComBat (cov) and our proposed CS-ComBat) varies when batch-class design is unbalanced, where ComBat tends to remove batch effects indiscriminately, while the reverse is true for CS-ComBat and ComBat (cov). This is attributable to the confounding between the group and batch factors and the strategies of performing ComBat (aware or not aware class information). Given a dataset with two classes and batches, and the batch-class design is unbalanced. Suppose ComBat is applied on the entire data with only specified batch factors, ComBat cannot identify those information associated with class factor; hence, it removes those class effects. Suppose ComBat (cov) applied to the data, besides batch factors included, class factors are also included as a covariate. At this time, the essence is to let the method retain the signal related to class factors. When batch-class confounding exists, the method cannot completely remove all batch effects, and at the same time, after batch effect correction, the part of the signal related to both batch and class factors may also remain. Suppose CS-ComBat applied to the data, where split the data by its class, and individually corrected by ComBat. When batch-class design is balanced, data presented to ComBat without class-associated variation, ComBat does a thorough job of removing all variation correlated with batch factors. However, when batch-class design is unbalanced, especially in the severely unbalanced case, data presented to ComBat with strong class-associated variation, CS-ComBat tends to preserve most class effects at the cost of not removing batch effects thoroughly.

When batch-class design is unbalanced, it can lead to problems for batch effects correction, since separating batch effects from real changes due to the phenotype (class effects), and avoid the risk of removing meaningful information is very challenging. However, when batch-class design is balanced, separating batch effects

from real changes due to the phenotype (class effects) is not a hard problem, then BECAs can perform well in this case. Hence, applying rebalanced approaches to make the data become balanced might help BECAs to correction batch effects when batch-class design is unbalanced. And this is verified in our results. Our results show that, SMOTE is a useful strategy in coordinated with BECAs to achieve good performance when batch-class design is unbalanced under medium CEP to high CEP. After applying SMOTE, the BECAs performance become quite similar and stable. Then either method could be chosen.

CS-ComBat with a very high Jaccard coefficient but with some decrease after SMOTE. A possible explanation might be that when batch-class design is unbalanced, and class effects smaller than batch effects, it is hard to separate the class effects from the batch effects, some batch effects confounding with class effects are preserved, then some false positives are inevitably introduced in the removing batch effects while try to preserving class effects as much as possible. Leading to a high Jaccard coefficient, which means the high Jaccard coefficient of CS-ComBat might be caused by both true positive and false positive when batch-class design is unbalanced and class effects smaller than batch effects. While after SMOTE, it is the optimal case for CS-ComBat to separate the class effects from the batch effects, then less false positive was made, Jaccard coefficient was then with some decreased, but it is still better than other approaches and the data without correction.

For the reason why we consider SMOTE but not other rebalancing approaches? Although there are several approaches that have been developed to overcome the class-unbalancing problem, either by under-sampling (dropping the majority samples to leading balanced class) or over-sampling (sampling from the minority class to create a balanced class). For under-sampling, it with the advantage of improving run time and storage troubles. However, its disadvantage is to lose important information by dropping samples from the majority class [130, 131]. For over-sampling, the advantage is that it does not cause information loss. The disadvantage is that it may lead to over-fitting as it oversampling the minority class [123, 130, 131]. For SMOTE, it generates artificial data instead of oversampling from the minority class to overcome imbalances problems. A powerful method goes beyond under-sampling and over-sampling. This approach without information loss and shows in improving performance compare with oversampling as SMOTE does not draw from a pool of

instances that is already small [123], then in this study, we consider the SMOTE approach. By applying rebalanced approaches: SMOTE to let sample class become balanced and to overcome the batch-class unbalanced problem, to make the BECAs performance become similar. Then either method can be chosen.

The application of CS-ComBat, ComBat (cov) and SMOTE are affected by the CEP. When the batch-class design is severely unbalanced, CS-ComBat and ComBat (cov) are most effective when CEP is high. ComBat (cov) and CS-ComBat can be used along to rescue the performance in the severely unbalanced case under medium CEP or high CEP. Also SMOTE plus different BECAs are recommended using when class-effect proportion ranges from medium to high.

Although here, we focus on methodology, this CS-ComBat strategy and SMOTE coordinate with BECAs will also have many practical uses in a real biological environment. Due to the scarcity and high cost of clinical samples, the size of the data samples obtained by researchers is limited and lacks power. Many researchers combine their data with data in public databases, such as Gene Expression Omnibus [24] and ArrayExpres [132], to increase statistical power. In this process, batch effects are inevitably introduced. Also, due to limited samples and budget, or improper experimental design, many data are inevitably unbalanced. Therefore, doing batch effect correction better will help us make good use of the biological data that has been invested in a lot of resources. In addition, the high CEP phenotype data exists in specific biomedical phenotypes, such as cancer. In this case, it is suitable for CS-ComBat.

In this study, we consider the gene expression data, data from other platforms (such as proteomics) are not included. As strong CEP and are also commonly exist in other platforms, the CS-ComBat, ComBat (cov) and rebalancing data with SMOTE in prior, then coordination with BECAs may also be used.

There are limitations though: although the results presented here provide evidence that CS-ComBat, ComBat (cov), and SMOTE coordination with BECAs are useful in preserve the class effect and also remove batch effects, this is only so in the context of how the batch effects are simulated here. While we used published procedures for simulating and detecting batch effects, we cannot guarantee that batch effects always manifest non-uniformity or that how we simulate batch effects is the only or correct way. We expect that similar criticisms may also be leveled in the way

differential features are generated as being simplistic, and that the precision-recall differential between the test BECAs are merely artifacts. In defense, by repeating hundreds of simulations, we are able to observe both convergence and conservation. We are therefore confident of the precision-recall differential between the tested BECAs. As to why CS-ComBat, ComBat (cov), SMOTE does not work well in when CEP is small, it is possible that give lower CEP, it is hard to select the relevant genes.

If the experiment is severely unbalanced, the commonly used BECAs (Ratio-A, Ratio-G, BMC) decline greatly in performance. SMOTE is a useful strategy to rescue the performance in this case under high CEP. SMOTE can make the performance of different BECAs becomes quite stable and similar, and they are so similar that it no longer matters what kind of methods is used. Alternative methods like ComBat (cov) and CS-Combat can also be used to rescue the performance in this case.

In summary, ComBat is wildly used for batch effects correction in -omics datasets. However, when batch-class design is unbalanced, especially in the severely unbalanced case. Blindly apply the commonly used ComBat strategy, which deploys ComBat on the whole data with only consider batch information is not suggested, and leads to overcorrect batch effects while losing class effects (“throwing out the baby with the bathwater”) as well as poorer statistical feature selection. Fortunately, there are alternative strategies can be used to overcome this issue. Here, we show that “CS-ComBat” and “ComBat (cov)” approaches not only readily outperform the blind approach to ComBat, they are also robust in preserving class effects when CEP is high. Besides, rebalancing data with SMOTE first following by different BECAs can also lead to better performance than doing batch effect correction on unbalanced data alone. Finally, taking account of removing batch effects is a hard problem, we proposed a comprehensive strategy to deal with batch effects in high-throughput technologies studies.

Chapter 4. Class-specific ComBat for correcting batch effects in high-throughput data with a focus on small sample size

4.1 Introduction

High-throughput technologies can reveal the expression profiles of thousands of genes in a sample at the same time, allowing people to obtain a complete picture of the entire gene, which has led scientists to use it for large-scale detection, leading to a shift into the paradigm of precision medicine. High-throughput technology has been applied for hunting disease-associated biomarkers[133, 134], clarifying biological function[2], understanding disease-related subtypes[135], elucidating disease etiology[136, 137], understanding disease diagnosis and prognosis[138, 139], inferring cancer-related mechanism[140, 141], characterizing immunity response[142, 143], and pharmacogenomics[144, 145]. However, it is challenging to directly use high-throughput raw data generated by high-throughput technology, because of the inherent data quality problems caused by batch effects, which may lead to misleading conclusions.

Batch effects impede the analysis of high-throughput data, when batch effects are mild, it may cause bias in the downstream functional analysis. When batch effects are strong, it leads to the following phenomena: some truly related genes are not detected in some samples (false negatives) and/or some unrelated genes are selected (false positives)[5, 17]. There are several reasons attributed to the cause of batch effects, including in the different labs, generated by different times, reagent lots, protocols, and sequenced by different platforms[12, 15, 99]. In order to obtain more reliable data from high-throughput technologies, it is necessary to correct the batch effects[146]. In addition, due to time, money, and sample limitations, many high-throughput data samples are relatively small.

Many methods have been developed to correct batch effects, such as Distance-Weighted Discrimination (DWD)[13], Batch mean-centering (BMC)[59], Ratio-based approaches[15], the Empirical Bayes (EB) method, ComBat[14] and Surrogate Variable Analysis (SVA)[60]. Among them, most batch effects correction algorithms

are not suitable for small samples [16]. One exception is ComBat, which is based on the Empirical Bayesian algorithm, robust against a small sample size[14]. In the previous chapter, we examined the limitations of ComBat and proposed a CS-ComBat strategy, and evaluated its performance against other ComBat strategies. But we focused on the larger datasets, while the small dataset was not investigated.

To fill this knowledge gap and to provide guidance in choosing a suitable method given the small sample size scenario, we evaluated our proposed CS-ComBat against another two strategies of performing ComBat on gene expression data with both real and simulated batch effects. First, to evaluate their ability to remove batch effects and preserving class effects, both qualitative and quantitative approaches are used. Next, to investigate their impact on inter-sampling similarities, the Jaccard coefficient was employed. Finally, to test if data integrity was compromised due to batch-effect removal, statistical feature selection was employed to evaluate the functional consequences of CS-ComBat and ComBat, ComBat (cov) on simulated datasets where we know the real features *a priori*.

4.2 Materials and methods

The design of this study is outlined in Fig. 4-1.

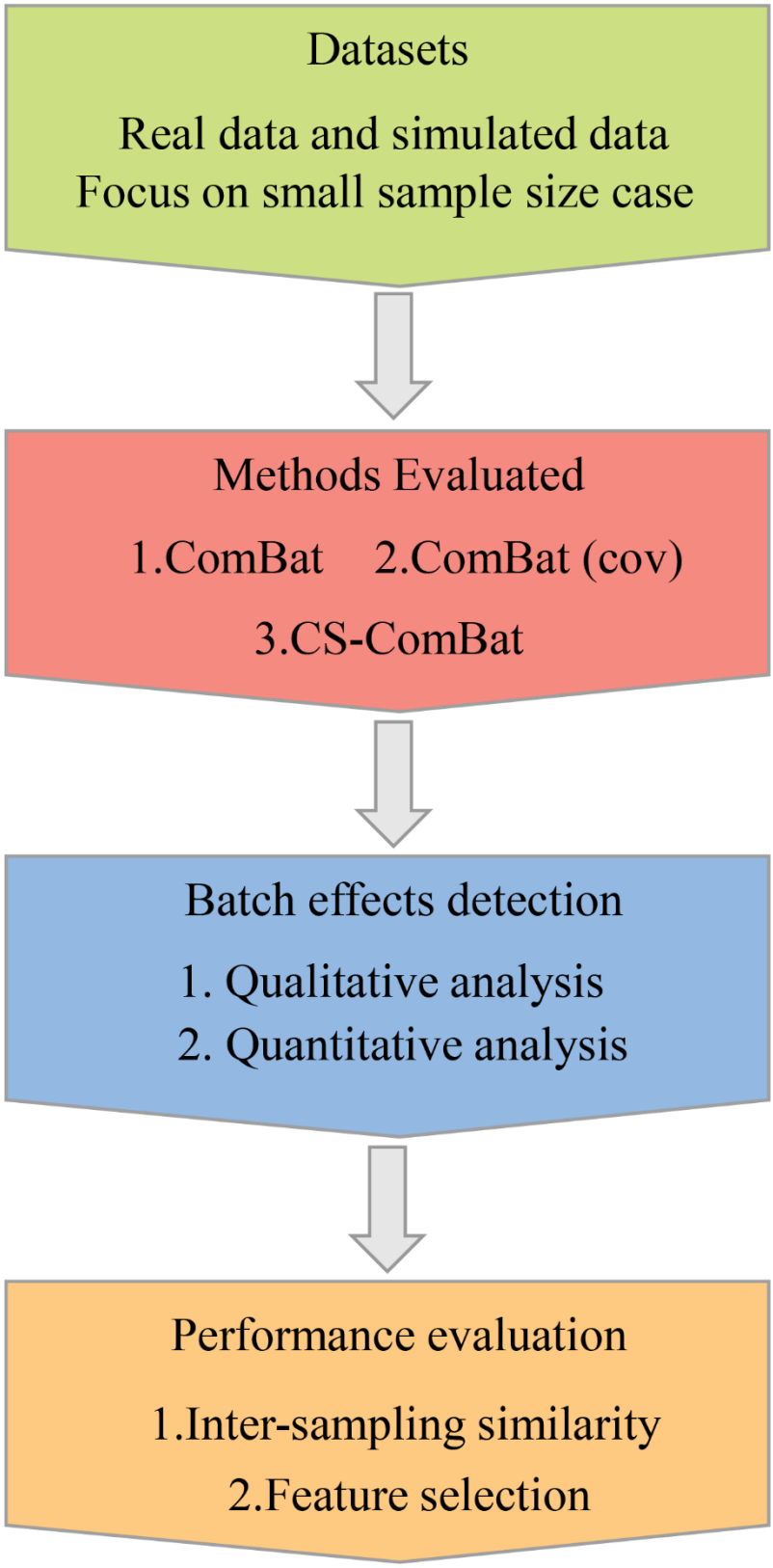


Fig. 4-1. Schematic overview of this work.

4.2.1 Real data

4.2.1.1 Non-Pregnant Mice (NPM) dataset: for creating natural batch effects

NPM dataset[147] includes 24 samples equally distributed across 4 classes (6 samples per class). Batch factors are distributed as 3 processing batches, with each batch contain 8 samples. This data is available in the HarmanData R package[62]. Since NPM has 3 processing batches and 4 groups, we can create any combination from samples derived from two processing batches and 2 different groups to simulate real batch effects and class effects. The procedure is described below:

Four samples from group 1, with 2 different batches (batch 1 and 2) were chosen as part 1. Four samples from group 2, with 2 different batches (batch 1 and 2) were chosen as part 2. Then part 1 and part 2 were merged to generate real class effects and batch effects. The merged dataset was designated NPM.1H and NPM.2H.

4.2.2 Simulated data

In order to determine the performance of ComBat, ComBat (cov) and CS-ComBat in the small sample size scenario, we simulated data based on a real data (GSE53334[106]) to mimic the real situation by using the "polyester" package[110]. Class and batch effects were incorporated by using Leek's method[76]. We simulated 100 expression studies and each simulated data with 8 samples and 1000 genes for the small sample size case, and batch-class design is balanced. Besides, we also consider their performance in the larger sample size scenario by increasing the sample size of the data from 8 to 80 gradually.

4.2.3 Data processing and other idiosyncrasies

For the real data, which has already pre-normalized, and was directly subjected to batch effects removal. As we do not know the true differential features in real data. We evaluate real data for batch-removal consistency and inter-sampling similarity. For the simulated data, which was first transformed using log followed by batch effect removal and statistical feature selection.

4.2.4 Batch effects correction algorithms (BECAs)

ComBat, ComBat (cov) and class-specific ComBat was performed as 3.2.4.

4.2.5 Batch effects detection methods

4.2.4.1 Qualitative approach: principal components analysis (PCA)

PCA coupled with scatterplot was used to detect batch effects, it was performed as 2.2.5.

4.2.4.2 Quantitative approach: partial redundancy analysis (pRDA)

pRDA was performed as 3.2.5.

4.2.6 Performance evaluation

4.2.6.1 Inter-sampling similarity

In the real data, inter-sampling similarity was evaluated with the Jaccard coefficient, which was calculated as 3.2.6.

4.2.6.2 Statistical feature selection

In order to assess the impact of CS-ComBat, ComBat and ComBat (cov) on data integrity, we used statistical feature selection followed by precision, recall and F-score measurement. These measurements were calculated as 3.2.7.

4.3 Results

4.3.1 On small datasets, CS-ComBat is capable of removing batch effects more thoroughly

We first employed the 2-dimensional (D) PCA scatterplots to visualize the class and batch effects. In 2D PCA scatterplots, PCs 1 and 2 present the individual samples, the shapes and colors represent class and batch, respectively. And the circular boundaries display the distributions of sample-colors, thus, if strong batch effects are present, they will separate out.

We tested the methods against datasets that contain real batch effects and simulated batch effects **(**Fig. 4-2**)**. For the data with real batch effects, 2D PCA scatterplots show that ComBat, ComBat (cov) and CS-ComBat successfully remove

batch effects as the samples clustered by class rather than batch. Among these, CS-ComBat seems to remove batch effects more thoroughly than other methods as batch 1 and batch 2 nearly completely merged together **(**Fig. 4-2A**)**. This result is reproducible in other tested dataset with real batch effects (Appendix D: Fig. D1). In the real data, we do not know the truth *in prior*, thus we further investigate the performance of 3 ComBat strategies in the simulated data. In the original data, 2D PCA scatterplots presented that it was clustered by class in PC1. Simulating batch effects results in two batches of each class in PCA, which destroys the class-specific clustering of the original data, this effect is most pronounced when color boundaries are separated, as can be observed in Fig. 4-2B**.** Similar as the observation in the real data, all the methods perform well in removing batch effects in the simulated data. Again, CS-ComBat appears to remove batch effects more thoroughly.

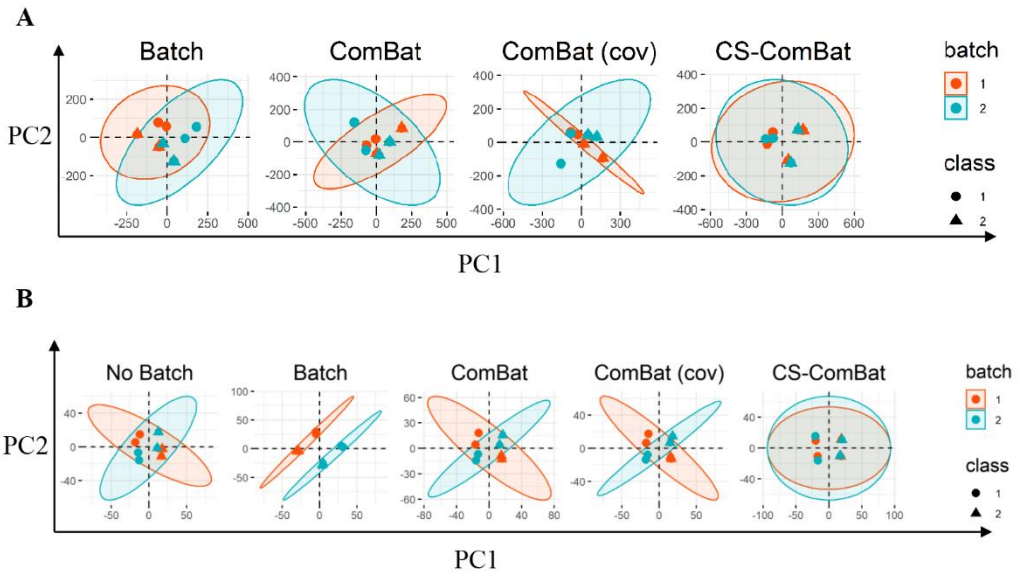


Fig. 4-2. Qualitative evaluation of three ComBat strategies using 2D-PCA scatterplot. **A**. data with real batch effects. **B.** data with simulated batch effects. ComBat, ComBat (cov) and CS-ComBat: batch correction with ComBat, ComBat (cov) and CS-ComBat.

2D PCA scatterplots give us a rapid inspection of the results, however, it just offers a crude approximation of batch effect correction efficiency. For a more rigorous assessment, quantitative methods are needed to evaluate the quality of batch effects correction. In this study, we consider the partial redundancy analysis (pRDA)

method. pRDA can be used to measure the percentage of variance in a dataset that can be attributed to batch and class before and after batch correction.

Fig. 4-3 shows the batch correction results, obtained using pRDA, in the datasets comprising real batch effects and simulated batch effects. For the data with real batch effects, after correction, all the tested methods can remove batch effects and preserve class effects (Fig. 4-3A). Amongst, CS-ComBat showed the best performance for batch effects correction and class effects preservation. For batch effects, CS-ComBat with a batch variance of 0.00, while ComBat (cov) and ComBat had a higher batch variance, with a value of 0.02 and 0.05, respectively. For the class effects, CS-ComBat with a class variance of 0.46, while ComBat (cov) and ComBat had a lower class variance, with a value of 0.31 and 0.25, respectively (with at least 1.48 fold difference). This result is reproducible in other data with real batch effects (Appendix D: Fig. D1), and also a similar finding was also observed in the data with simulated batch effects (Fig. 4-3B).

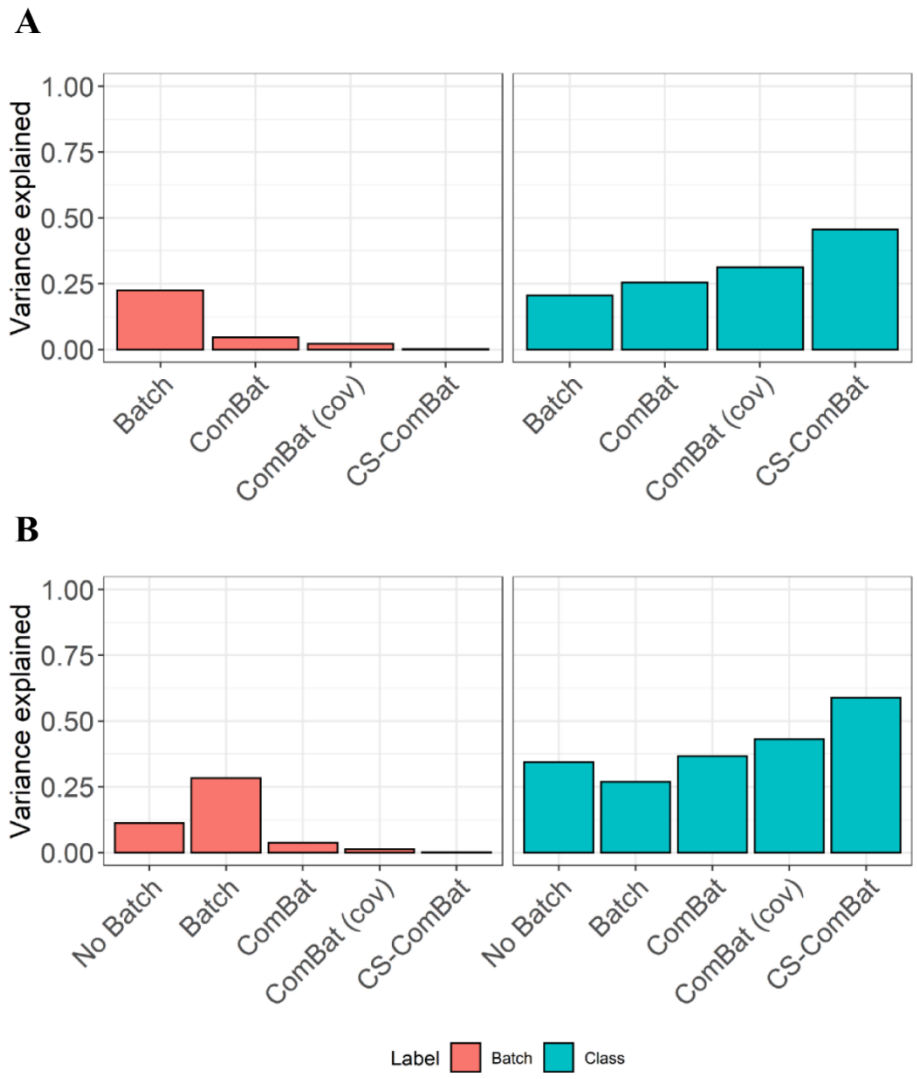


Fig. 4-3. Quantitative evaluation of three ComBat strategies using the pRDA. **A**. data with real batch effects. **B.** data with simulated batch effects. ComBat, ComBat (cov) and CS-ComBat: batch correction with ComBat, ComBat (cov) and CS-ComBat.

The results in Fig. 4-3B only include results based on one simulation. For evaluating the reproducibility of the results, we employ pRDA across 100 simulated datasets. Fig. 4-4 shows that after applying ComBat, ComBat (cov) and CS-ComBat to correct batch effects, CS-ComBat was consistently perform best in removing batch effects while preserving class effects under small sample size scenarios. Data that has been batch corrected via CS-ComBat has the lowest batch variance but highest class variance. pRDA method facilitates rapid batch effects detection and quantitation across many simulations, and the results are stable.

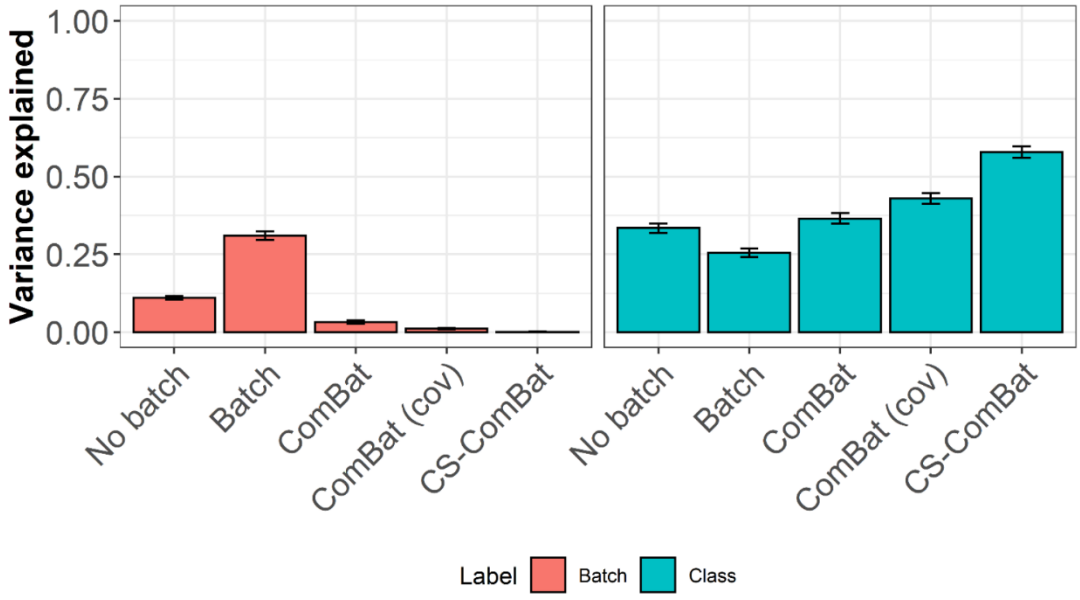


Fig. 4-4. Batch effects detection with pRDA in the data with simulated effects (100 simulations). No Batch: data with only class effects but not batch effects. Batch: data with simulated batch effects. ComBat, ComBat (cov) and CS-ComBat: batch correction with ComBat, ComBat (cov) and CS-ComBat. Results represent mean ± SD.

4.3.2 On small datasets, CS-ComBat results in higher inter-sampling similarity

In order to investigate the impact of batch effects correction on the data with real batch effects, we examine inter-sampling similarities through the Jaccard coefficient[58]. Fig. 4-5 shows that all the tested BECAs can results in a higher Jaccard coefficient compared to the batch case, indicating an improvement of inter-sampling similarities after batch effects correction. Amongst, the CS-ComBat strategy generally performs the best (in terms of Jaccard coefficient) in inter-sampling similarities, with a median Jaccard coefficient of 0.62, followed by ComBat (cov) and ComBat with a median Jaccard coefficient of 0.42 and 0.24. And this results is consistently reproducible in other data with real batch effects (Appendix D: Fig. D2).

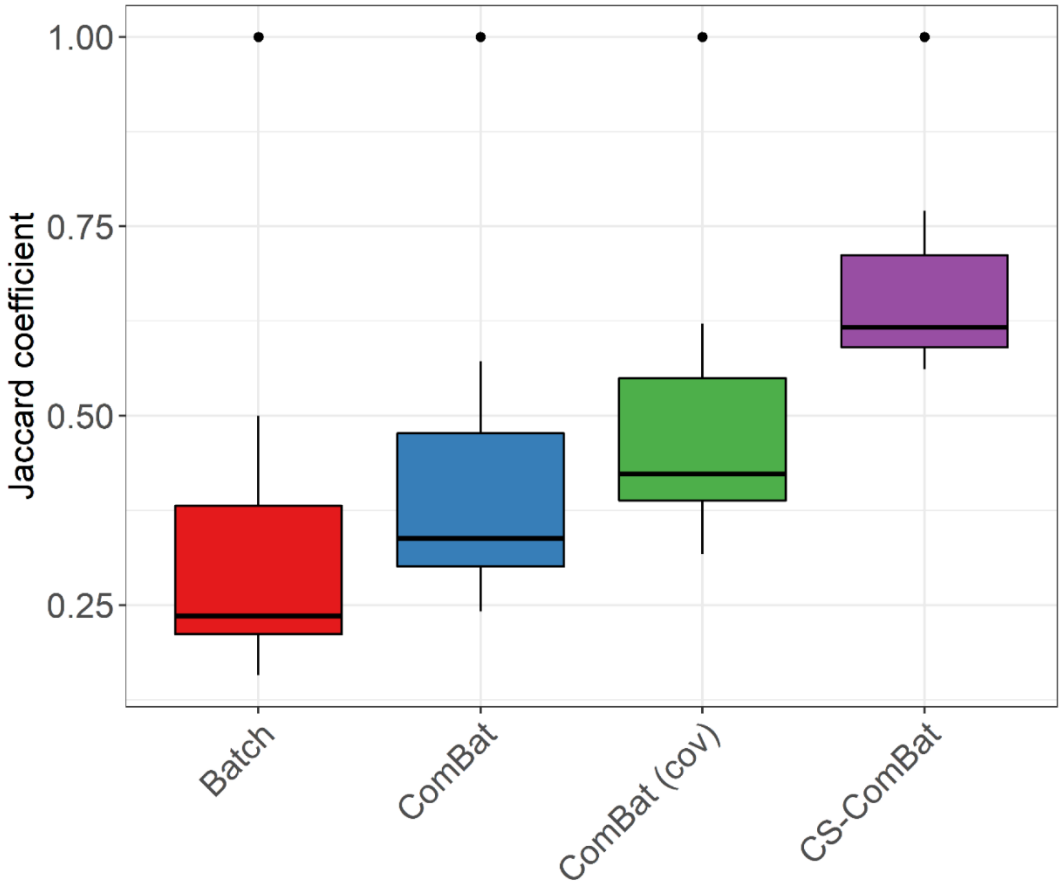


Fig. 4-5. Inter-sampling similarity evaluation with Jaccard coefficient in data with real batch effects (NPM. 1H). Batch: data with real batch effects. ComBat, ComBat (cov) and CS-ComBat: batch correction with ComBat, ComBat (cov) and CS-ComBat.

4.3.3 On small datasets, CS-ComBat produces higher recall

Resolving batch effects is not an easy task since batch effects are complex and may not completely be removed, and batch effects correction may have important implications on the downstream feature selection. On real data, we do not know the true differential expression genes in *a priori*. Thus we further investigate the performance of ComBat, ComBat (cov) and CS-ComBat on simulated data where the differential expression genes are known in advance. We test the impact of batch effects correction with ComBat, ComBat (cov) and CS-ComBat on data integrity by considering the feature selection following with precision, recall and F-score. Fig. 4-6 shows feature selection results of data with simulated effects, after applying different

BECAs to correct batch effects, CS-ComBat results in lowest precision but highest recall compared to ComBat and ComBat (cov) during feature selection.

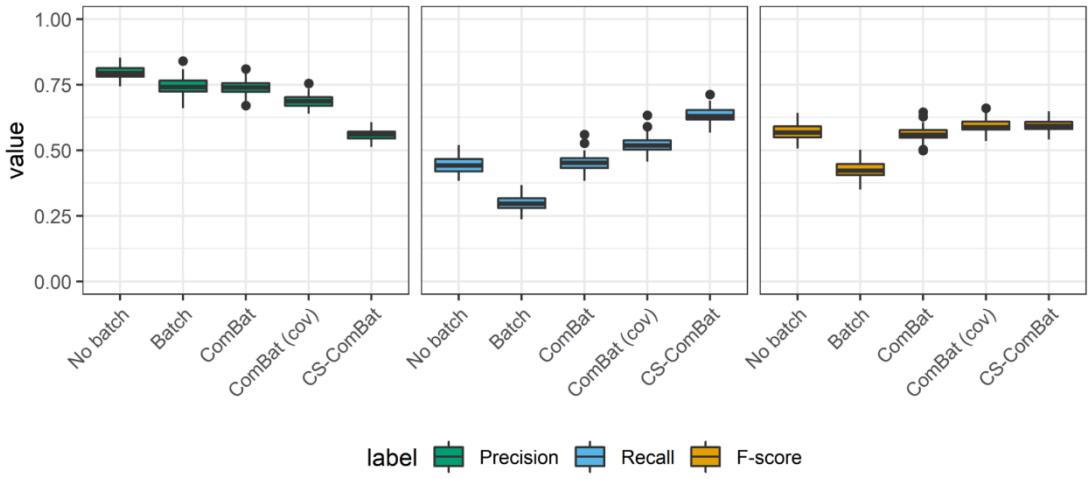


Fig. 4-6. Feature selection results of data with simulated effects (100 simulations). From left to right indicate the distributions of precision, recall and F-score, respectively. No Batch: data with only class effects but not batch effects. Batch: data with simulated batch effects. ComBat, ComBat (cov) and CS-ComBat: batch correction with ComBat, ComBat (cov) and CS-ComBat.

4.3.4 On larger datasets, the performance of ComBat，ComBat (cov) and CS-ComBat become merged together

The previous results are tested in datasets with small sample size (n = 8), to test the performance of CS-ComBat when the sample size becomes larger. We increased the sample size of data gradually from 8 to 80. In the larger sample size scenario, pRDA results show that all tested BECAs remove batch effects thoroughly and preserve class effects well, it appears that batch variance of ComBat, ComBat (cov) and CS-ComBat was close to each other, similar for the class variance. And this result is consistent across 100 simulations (Fig. 4-7A). Next, we also test the impact on data integrity based on the feature selection. It appears that feature selection results of CS-ComBat, ComBat and ComBat (cov) become similar and there is no clear winner given the F-score when the sample size becomes larger (Fig. 4-7B). Collectively, these results suggest that ComBat, ComBat (cov) and CS-ComBat performance converges in the larger sample size scenario.

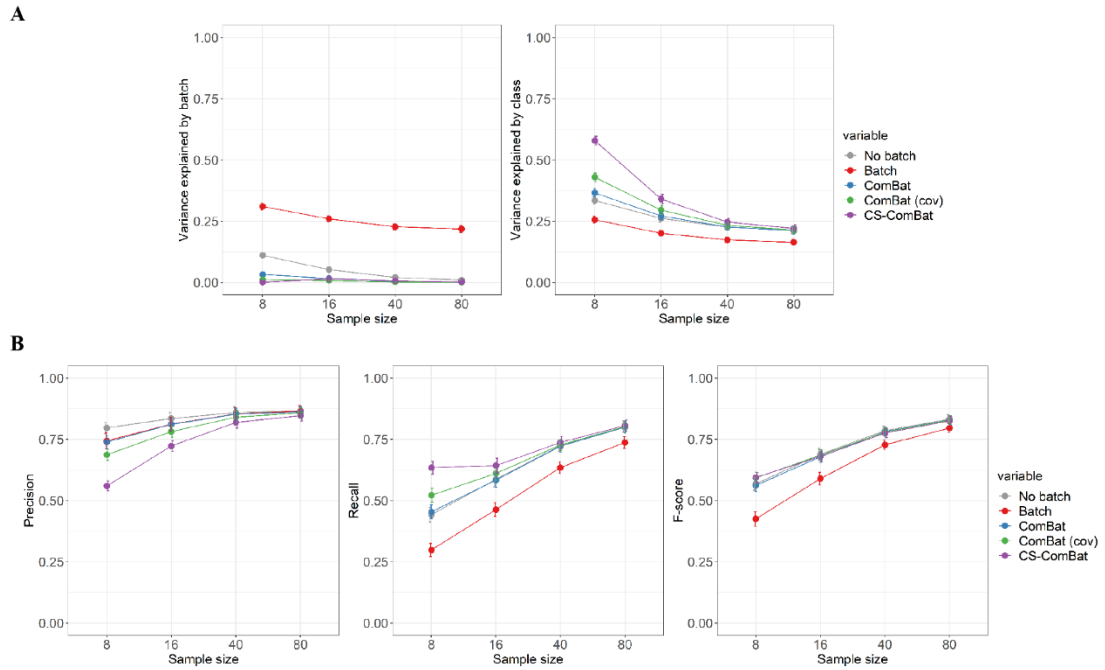


Fig. 4-7. pRDA and feature selection results of data (increase sample size from 8 to 80) with simulated effects (100 simulations). **A**. variance explained by batch (left) and class (right) calculated with pRDA. B. From left to right indicate the precision, recall and F-score, respectively. Error bar represents standard error of 100 simulations.

4.4 Discussion and conclusions

Based on pRDA and other evaluation of batch effects, CS-ComBat is better at removing batch effects across all evaluations, including simulated and real data under small sample size scenarios. This performance gain is also data-independent.

While this may be true, removal of batch effects does not necessarily equate to performance gains in statistical feature selection. It turns out that CS-ComBat corrected data produce higher recall, but lower precision compared to ComBat and ComBat (cov) corrected data. Both metrics are important and different analytical needs: favoring high precision means getting only the right answers and making few errors. But it usually comes at the cost of recall, which stymies obtaining enough variables (genes or proteins) to build sufficiently robust models. Conversely, favoring high recall means getting most of the relevant variables. But it comes at the cost of precision, meaning false positives will also be selected.

When to use either ComBat or CS-ComBat depends on analytical need: when we want to design a drug target, we only need one protein to target. So, we would favor high precision (use ComBat). When we want to understand mechanism, we need as many relevant genes as possible. Too few means we cannot build a complete model. So, in this case, recall is favored (use CS-ComBat).

As to why removal of more batch effect results in loss of precision but gains in recall, it is possible that retaining the component of batch effect correlated with class shifts *P*-value distributions down for features with strong effect sizes (making it easier to detect these), but drives up the p-values for features with weaker effect sizes (making it harder to detect these) as well as those features that are irrelevant (thus a lower false positive rate). As to why increasing the sampling size results in the performance of ComBat, ComBat (cov) and CS-ComBat becomes merged, which may due to the statistical power was increased when the sample size becomes larger.

Batch effects confound real signal detection in biological data. Although ComBat has been shown to be a superior batch-effect correction method, its modus operandi involves direct presentation of multi-class data with only the batch effects specified. We demonstrated that this procedure results in incomplete removal of batch effects when sample size is small, and our proposed alternative procedure CS-ComBat, can remove batch effects more thoroughly. Unfortunately, more thorough removal of batch effects results in tradeoffs for downstream statistical feature selection. CS-ComBat is good at recall, but has lower precision than native ComBat. Hence, the decision when to use which should be dictated more by analytical need, than the need to thoroughly eradicate batch effects.

Chapter 5. Batch effects correction on proteomics data: correcting at peptide level or protein level?

5.1 Introduction

Proteomics is the large-scale study of proteins that enable us to simultaneously monitor protein expressions at a proteomic scale[148-152]. Proteomics are used to study a series of biological systems at the protein level[151, 153-156]. In the past few years, with the advancement of technology, for example, high-performance protein extraction procedures (PCT)[157] and sequential windowed acquisition of all theoretical mass spectra (SWATH-MS)[158], these advanced technologies have made the application of proteomics become more and more extensive[159-162]. It can be used to fully understand the changes of proteins in diseases and treatments, which reveals their usefulness in a wide range of biological disciplines, including bio-markers detection[163, 164], cancer classification[160, 165], potential drug target[166, 167], investigating biological mechanisms[168-170] and clinical diagnostics[171-173]. Unfortunately, the application process of proteomics is plagued by batch effects[5, 12].

Proteomics data contain batch effects due to various technological limitations. In the bottom-up proteomics, several steps are attributed to the cause of batch effects: (1) samples acquisition: different samples may be stored in different places; (2) Samples preparation: samples may be handled by different personal or different reagent lots, and (3) the measurement of mass spectrometry: the number of samples that can be measured in one batch was limited. The batch effects in proteomics data would significantly reduce the performance of downstream analysis, such as the detection of differentially expressed proteins and protein clustering. Therefore, batch effects correction has become an important component of proteomics data analysis.

Batch effects correction algorithms (BECAs) used in proteomics are borrow from those BECAs developed for gene expression studies. There are various algorithms have been developed for batch effects correction. These include surrogate variable analysis (SVA)[60], an Empirical Bayes approach called ComBat[14], batch

mean-centering (BMC)[59], ratio-based approaches[15], remove unwanted variation (RUV)[61], and a principal component analysis based approaches called Harman[62]. Among those methods, ComBat dominates for its outstanding performance in removing batch effects[99]. Whereas many studies focus on correcting batch effects at the protein level[12, 174, 175], the analysis at earlier peptide-level is seldom studied, and it is unclear that whether batch effects correction done earlier in the peptide level, would the final protein level be more correct? In addition, under the peptide level, there are ambiguous peptides exist, then what is the role of ambiguous peptides in batch effects correction? If we consider the ambiguous peptide, would the final protein level be more accurate?

To address these knowledge gaps, we have performed a systematic evaluation of batch effects correction at peptide level and protein level on proteomics data originated from advanced proteomics technology (SWATH). Under peptide level, we also investigate batch effects correction done at the ambiguous peptide level and unique peptide level. The evaluation was performed on datasets with both real batch effects and simulated batch effects, the methods were evaluated based on their ability to correctly remove the batch effects and preserve the class effects as well as on their impact on the data integrity by comparing their inter-sampling similarity and statistical feature selection metrics (precision, recall and F-score). Our results showed that the batch effects correction at peptide level performs as well as batch effects correction at protein level in terms of principal component analysis (PCA) and partial redundancy analysis (pRDA), inter-sampling similarity and statistical feature selection. This led us to provide practical guidance for batch effect correction in proteomics data.

5.2 Materials and methods

The analysis design is outlined in Fig. 5-1.

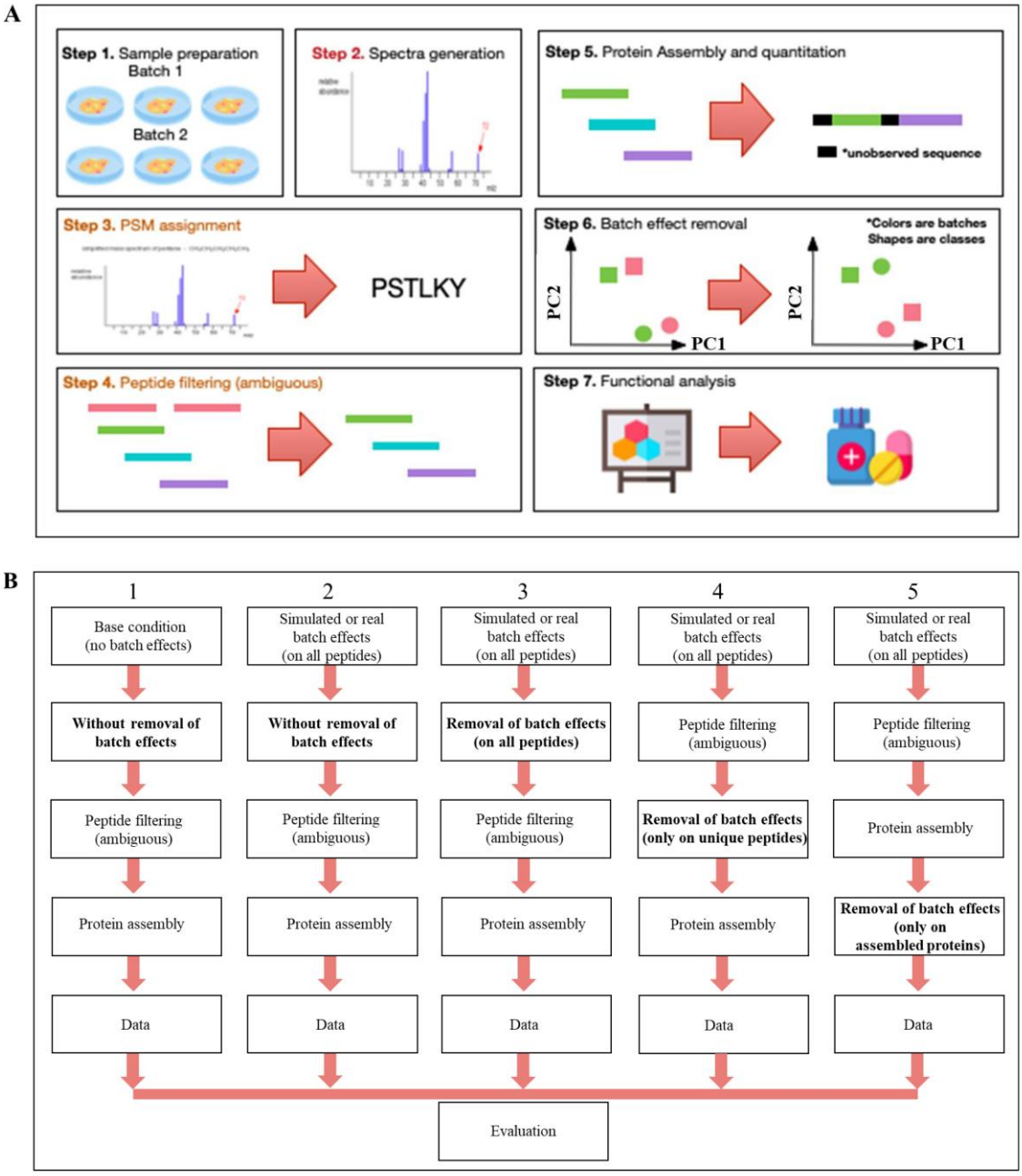


Fig. 5-1. Schematic overview of this study. **A.** Summary of the main steps in the common proteomics study from sample preparation to function analysis**. B.** Design pipeline from dataset to evaluation. For the peptide filtering, those peptides that are ambiguous (>1) were removed. For the protein assembly, we were taking the average of the peptides that map to each protein and protein supported by at least 1 unique peptide.

5.2.1 Datasets

The evaluation was performed in two publicly available proteomics datasets, renal cancer (RC) dataset[157] and renal cancer control (RCC) dataset[157], and these datasets provided the peptide list and protein list. Data information was summarized in Table 5-1. In brief, RC contained 24 samples, 2 classes (each class with 12 samples), 6 biological replicates, and 2 technical replicates (batches). The number of identified peptides, unambiguous peptides and ambiguous peptides are 32421, 25532, and 6889, respectively. The identification of proteins is through using the OpenSWATH against a spectral library (which includes 49959 reference spectra of 41542 proteotypic peptides derived from 4624 reviewed SwissProt proteins)[176]. Finally, the data with 3123 proteins given false-discovery rate (FDR) at 0.01. RCC contains 12 samples, with 1 class, 4 biological replicates, and 3 technical replicates. The number of identified peptides, unambiguous peptides, and ambiguous peptides are 27383, 21017, and 6366, respectively. Proteins identification was using the same approach (OpenSWATH) against the library as RC. Finally, the RCC data contain 2331 proteins given FDR at 0.01.

Table 1. The information about the data used in this study

Data	The number of identified proteins	The number of identified peptides	The number of unambiguous peptides	The number of ambiguous peptides	Sample sizes (classes* biological replicates *technical replicates)	Origins	Experiment type	Acquisition mode	Reference
RC (Renal cancer)	3123	32421	25532	6889	24 (2*6*2)	normal kidney tissue and clear-cell renal carcinoma	SWATH	Data-Independent Acquisition (DIA)	[157]
RCC (Renal cancer control)	2331	27383	21017	6366	12 (1*4*3)	human normal kidney tissue	SWATH	Data-Independent Acquisition (DIA)	[157]

5.2.2 Strategies of class and batch effects simulation and real batch effects generation

In RC data, although there are two batches that exist, the batch effects are very small as detected by partial redundancy analysis (pRDA) as shown in Appendix E: Fig. E1, then we treat the raw data as batch-free data, and introduce batch effects to batch 2 by using Langley's approach[111]. In particular, by splitting each class of dataset into two equal batches, technical repeat 1 and technical repeat 2, then we draw from one of five possibilities or p (20%, 50%, 80%, 100% and 200%) and insert batch effects for each peptide in technical repeat 2. This is expressed as[12]:

$$SC_{ij}' = SC_{ij} * (1 + p)$$

Where SC_{ij} and SC_{ij}' represent the primary and simulated spectral count from the jth sample of peptide i.

In RCC, there are 3 batches and 1 class, we can create three combinations by merging samples derived from two batches. the batch effects detection results of each combinations was shown in Appendix E: Fig. E1. Among these combinations, the data with the smallest batch effects was treated as batch-free data, then simulating both class effects and batch effects by using Langley's approach[111]. In particular, for the class effects simulation: In the seed dataset, half the samples are arbitrarily split into classes 1 and 2 (All simulated datasets follow this same split). In each simulated dataset, 20% peptides are randomly assigned as differential, with a simulated effect size drawn from one of p (20%, 50%, 80%, 100% and 200%) and applied on each sample in class 2.

We use the same method to simulate batch effects as above, by splitting each class of dataset into two equal batches, technical repeat 1 and technical repeat 2, then we draw from p randomly and insert batch effects for each peptide in technical repeat 2.

The data with the medium batch effects was not adopted. While the data with the largest batch effects was treated as natural batch effects. A summary of these data presented in Fig. 5-2.

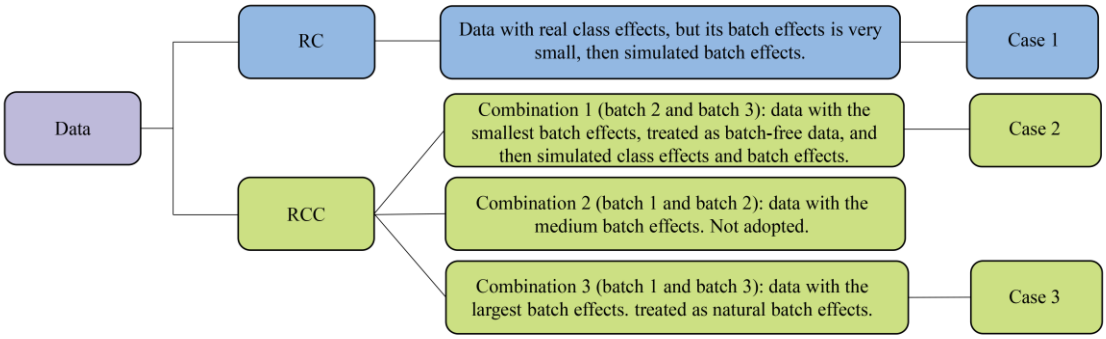


Fig. 5-2. A summary of data. Case 1: data with real class effects and simulated batch effects (based on RC data). Case 2: data with simulated class effects and simulated batch effects (based on RCC data). Case 3: data with a single class and real batch effects (based on RCC data).

5.2.3 Data pre-processing

Before batch effects simulation or generating real batch effects, missing value imputation was used to overcome the data holes in the proteomics data at the earlier peptide level. In particular, those data holes (NA values) in the peptide files were replaced with the mean value.

5.2.4 Batch effects correction algorithms (BECAs)

Batch effects correction with ComBat was performed as 2.2.4.

5.2.5 Batch effect detection approaches

5.2.5.1 Principal components analysis (PCA)

PCA was performed as 2.2.5.

5.2.5.2 Partial redundancy analysis (pRDA)

pRDA was performed as 3.2.5.

5.2.6 Performance evaluation

5.2.6.1 Inter-sampling similarity analysis

The inter-sampling similarity was performed as 3.2.6.

5.2.6.2 Statistical feature selection analysis

Statistical feature selection analysis was performed as 3.2.7

5.3 Results

5.3.1 Batch effects correction at peptide level or protein level can remove batch effects well and allow samples cluster by class rather than batch

PCA scatterplot was employed to evaluate batch effects and class effects before and after batch effects correction at different levels (Fig.5-3), if strong batch effects existed, it will separate out. We will discuss findings based one data, and then see if similar results are also observed in other tested data. As presented in Fig.5-3A, in the RC data without simulating batch effects, the PCA scatterplots show that the samples are separated by class, indicating class effects are dominated here, but batch effects are very small. After batch effects simulation, the samples are separated by batch, indicating batch effects exist. From those batch effects correction results. Batch effects correction at peptide level or at protein level can make the data samples separated by class rather than batch. And their sample distribution close to each other. Indicating Batch effects correction at peptide level or protein level remove batch effects well and perform similarly. These results are further validated in another dataset with simulated class and batch effects (Fig.5-3B) and also the data with real batch effects (Fig.5-3C). Besides view by PCA scatterplots, we also include the quantitative approach to detect the batch effects.

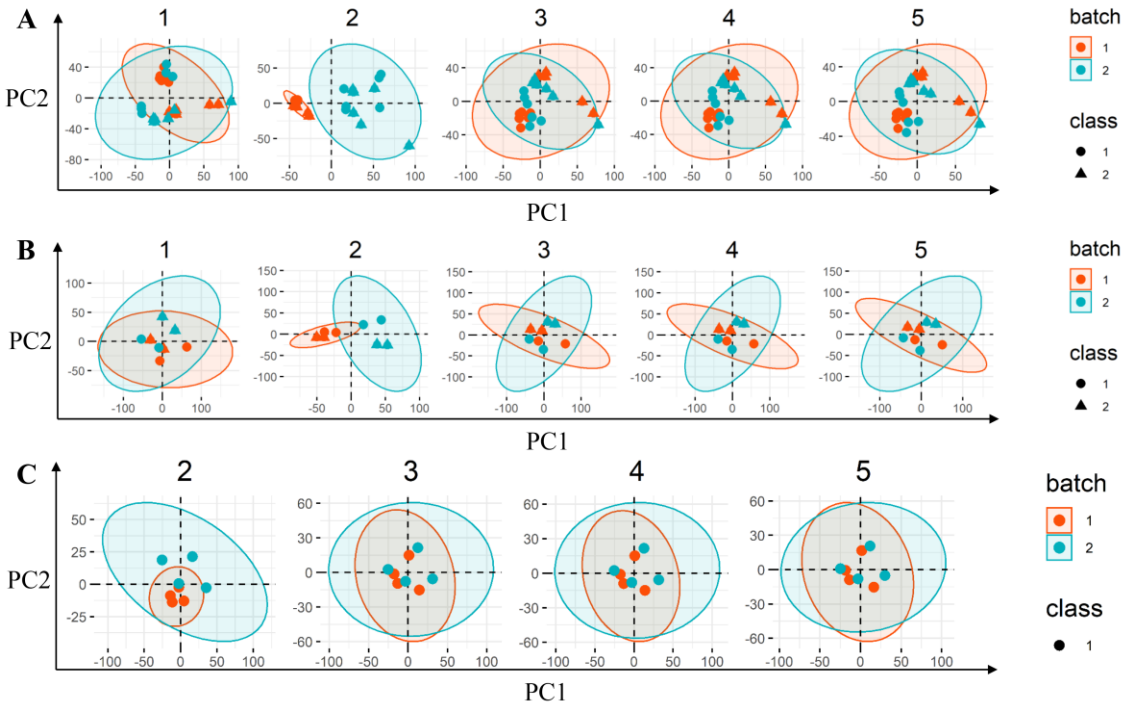


Fig. 5-3. Batch effects detection with PCA scatterplot in final assembly protein level. **A**. Case 1: data with real class effects, simulated batch effects (based on RC data). **B**. Case 2: data with simulated class effects and simulated batch effects (based on RCC data). **C.** Case 3: data with a single class, real batch effects (based on RCC data). 1: Base condition (no batch effects); 2: Simulated batch effects (on all peptides) in case 1 and case 2, while real batch effects (on all peptides) in case 3; 3: Removal of batch effects (on all peptides) by ComBat; 4: Removal of batch effects (only on unique peptides) by ComBat; 5: Removal of batch effects (only on assembled proteins) by ComBat.

5.3.2 Batch effects correction at peptide level or protein level can remove batch effects to a similar level in terms of pRDA

pRDA was used to show the variance explained by batch or class effects after batch effects correction at different levels. Ideally, after batch effects correction, the variance explained by class should be dominated and the batch effects expected be lower than class effects or even disappear. As shown in Fig. 5-4, after correction with ComBat, batch effects are lower than class effects indicating a good batch effects correction and class effects preservation, and batch effects correction at different level

perform similarly. They had a batch variance around 0.02, and class variance around 0.28. When considering data with simulated class and batch effects (case 2) or data with real batch effects (case 3), these observations are consistently supported, except in case 3 we ca not observe the class effects as the data with only one single class. Previously results based on one example, to test the consistency of the results, we apply pRDA across 100 simulated 100 data. Fig. 5-5 shows that after employ ComBat to correct batch effects correction at different level, the batch variance was consistently decreased while class variance was consistently increased, batch effects correction at different level with similar performance and the results are largely stable (Fig. 5-5).

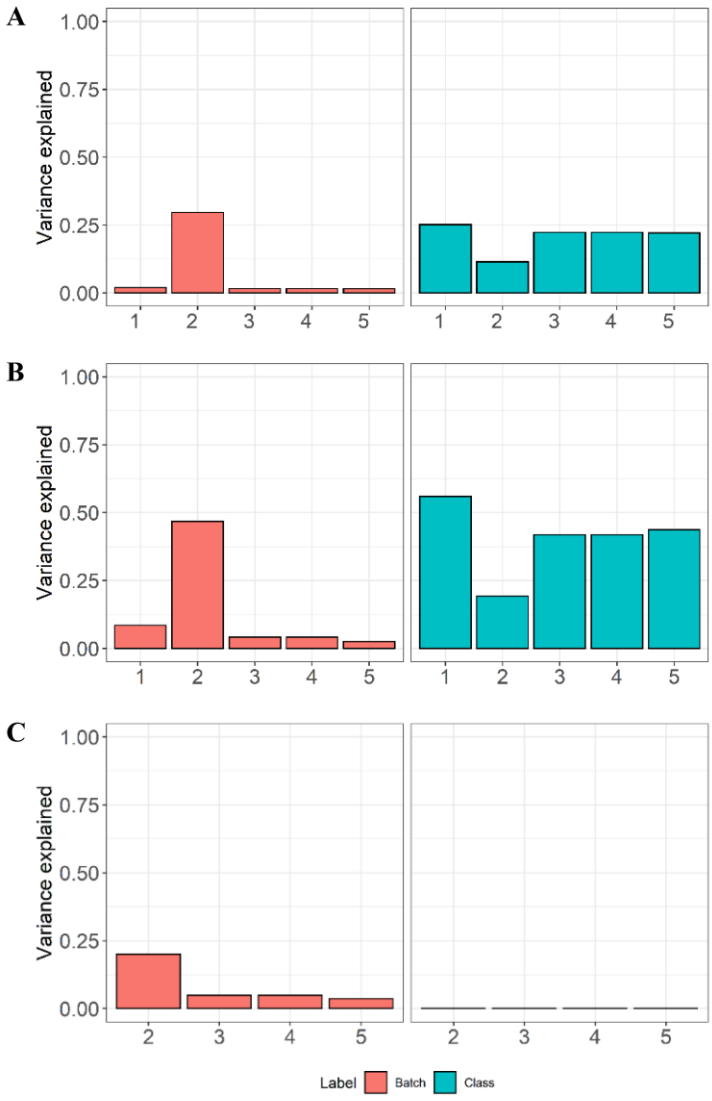


Fig. 5-4. Batch effects detection with pRDA in final assembly protein level. **A**. Case 1: data with real class effects, simulated batch effects (based on RC data). **B**. Case 2: data with simulated class effects and simulated batch effects (based on RCC data). **C.** Case 3: data with a single class, real batch effects (based on RCC data). X-axis indicate different variable. 1: Base condition (no batch effects); 2: Simulated batch effects (on all peptides) in case 1 and case 2, while real batch effects (on all peptides) in case 3; 3: Removal of batch effects (on all peptides) by ComBat; 4: Removal of batch effects (only on unique peptides) by ComBat; 5: Removal of batch effects (only on assembled proteins) by ComBat. Y-axis shows the value of variance explained by batch (red) or class (blue).

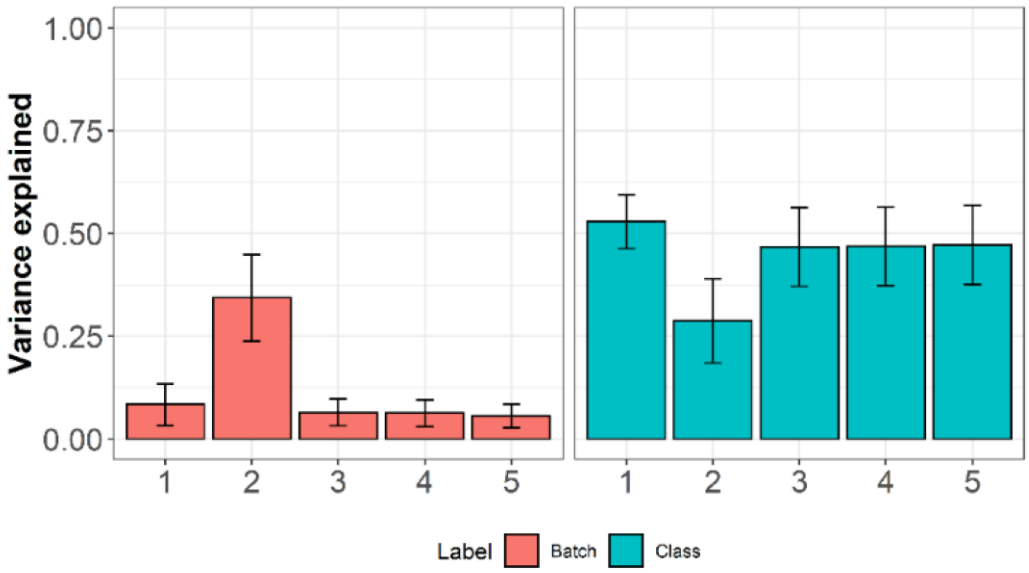


Fig. 5-5. Batch effects detection with pRDA (100 simulation data) in the final assembly protein level of Case 2: data with simulated class effects and simulated batch effects (based on RCC data). X-axis indicates different variables. 1: Base condition (no batch effects); 2: Simulated batch effects (on all peptides); 3: Removal of batch effects (on all peptides) by ComBat; 4: Removal of batch effects (only on unique peptides) by ComBat; 5: Removal of batch effects (only on assembled proteins) by ComBat. Y-axis shows the value of variance explained by batch (red) or class (blue). Results represent mean ± SD.

5.3.3 Batch effects correction at peptide level or protein level lead to similar inter-sampling similarity

In order to understand the impact of batch effect correction on the inter-sampling similarity of data, Jaccard coefficient was incorporated. As shown in Fig. 5-6, in the raw data, as expected, Jaccard coefficient is high, after simulating batch effects, the Jaccard coefficient decreased a lot, indicating a strong difference existed between the data simulated with batch effects and raw data and batch effects decreasing the inter-sampling similarity of the data. After batch effects correction, Jaccard coefficient was increased compared to the batch case suggests batch effects correction results in improving inter-sampling similarity. Among those tested scenarios, the results suggest that batch effects correction at peptide level (either in the all peptide

level or unique peptide level) perform as well as batch effects correction at protein level in terms of Jaccard coefficient, their distribution and median value are very close to each other (with median Jaccard coefficient around 0.63).

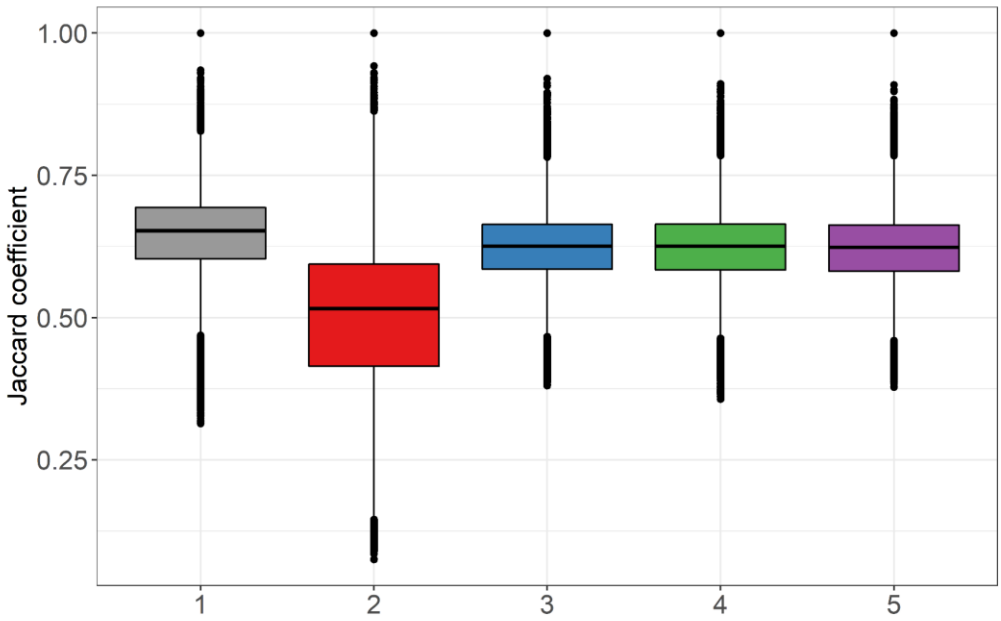


Fig. 5-6. Perform evaluation with Jaccard coefficient in the final assembly protein level of Case 1 (data with real class effects and simulated batch effects, data generated from RC data). X-axis indicates different variables. 1: Base condition (no batch effects); 2: Simulated batch effects (on all peptides); 3: Removal of batch effects (on all peptides) by ComBat; 4: Removal of batch effects (only on unique peptides) by ComBat; 5: Removal of batch effects (only on assembled proteins) by ComBat. Y-axis shows the value of Jaccard coefficient.

5.3.4 Batch effects correction at peptide level or protein level perform similar in terms of precision, recall and F-score

Batch effects can lead to inaccurate downstream analysis. Therefore, batch effects correction should not only be able to remove the batch effects satisfactory but also need to recover statistical feature selection performance back to their original state. Here, the precision, recall and F-score were calculated before and after batch effects correction at different levels. Results are shown in Fig. 5-6. In the raw data, as

expected, with good precision, recall and F-score, after simulating batch effects, the recall and F-score decrease a lot compare with raw data. Correction batch effects at different levels result in increasing the performance of recall and F-score compared to the batch case. And batch effects correction at peptide level (either in the all peptide level or unique peptide level) perform as well as batch effects correction at protein level in terms of precision, recall and F-score. Their precision, recall and F-score value are very close to each other.

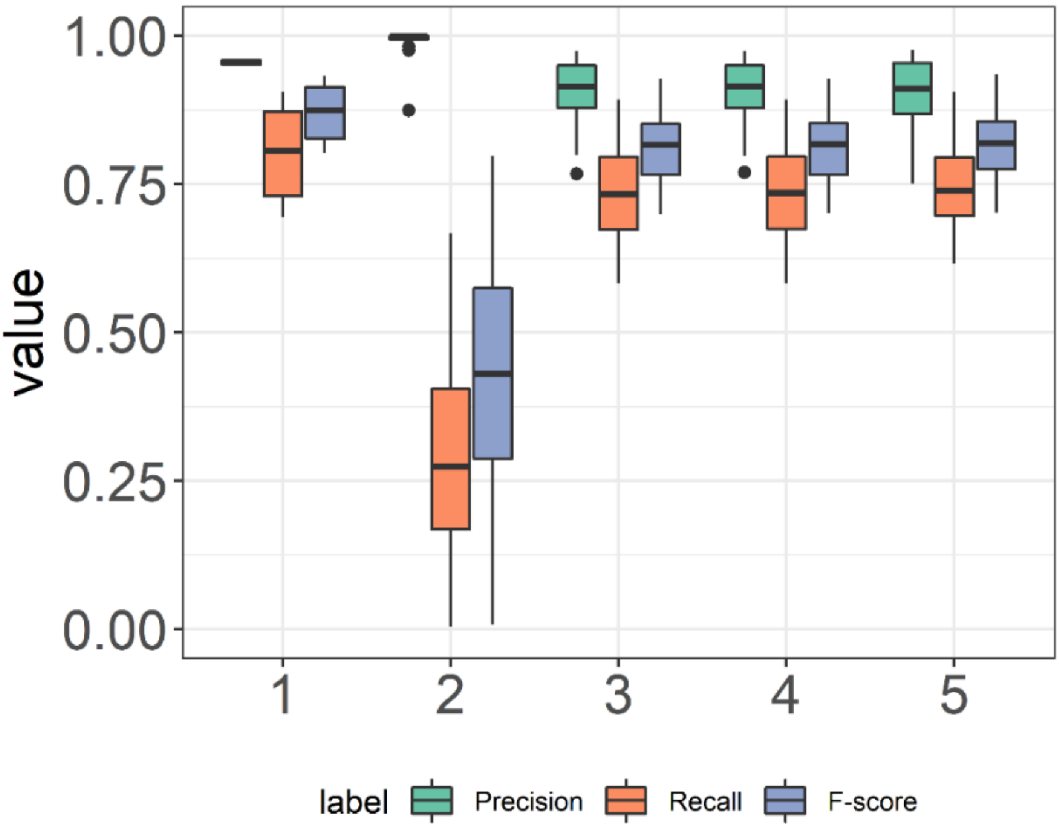


Fig. 5-6. Performance evaluation with feature selection in the final assembly protein level of Case 2 (data with simulated class effects and batch effects, data generated from RCC data). X-axis indicates different variables. 1: Base condition (no batch effects); 2: Simulated batch effects (on all peptides); 3: Removal of batch effects (on all peptides) by ComBat; 4: Removal of batch effects (only on unique peptides) by ComBat; 5: Removal of batch effects (only on assembled proteins) by ComBat. Y-axis shows the value of precision, recall and F-score.

5.4 Discussion and conclusion

According to the qualitative and quantitative evaluation of batch effects, batch effects correction at peptide level or protein level can remove batch effects well and preserve class effects, and also gain good inter-sampling similarity and statistical feature selection results. Overall, these findings suggest that batch effects correction at peptide level (either in the all peptide level or unique peptide level) performs as well as batch effects correction at protein level. Both lead to an accurate final protein level.

It seems that ambiguous peptides do not play a role in batch effects correction. As batch effects correction at all peptide level (where both ambiguous peptides and unique peptides existed) or at unique peptide level (where ambiguous peptides have been removed, only unique peptides existed), both lead to an accurate and similar performance in the final protein level.

When to use either batch effects correction at peptide level or batch effects correction at protein level rely on analytical need: if we aim to investigate splice variants, classify Isoforms and study post-translational modification, batch effects correction at protein level is too late, thus, we would do batch effects correction at peptide level. If we aim to investigate differential expression proteins, understand protein pathways and study protein-protein interaction, then correction at either peptide level or protein level can be used.

There are limitations in this study. Although we evaluated batch effects correction at different level based on the real proteomics data, this study is limited to the proteomics data by using the DIA mode, assume acquisition mode bias can be negligible, we expect these principles should be applicable to the proteomics data based on data-dependent acquisition (DDA) mode.

Although we using the published method to simulating batch effects and validate the results by using the data with real batch effects, we can not assure the way we simulated batch effects is the only or right way, and our simulated data and real data may not cover the full heterogeneity of the data in the proteomics field. More data could be included in future studies as a supplement.

Overall. This study investigated the correction of batch effects at different levels in the proteomics data, and evaluated its impact on data integrity and inter-sampling

similarity. The research results provide theoretical support for the correction of batch effects in proteomics. Meanwhile, correcting batch effects at different levels based on different aims has broad scientific significance for proteomics research.

Chapter 6. Conclusions, innovations and prospects

6.1 Conclusions

In this thesis, we have studied viable strategies for combating batch effects in high-throughput data. We clarified the practical limitations of batch effect correction algorithms and provided feasible batch effect correction strategies for this field. The main conclusions of this thesis are as follows:

1. BECAs are largely unaffected by upstream normalization procedures. While conventional normalization methods do not deal with batch effects at all, and they also do not affect downstream statistical feature selection. In less ideal situations where there are moderate levels of confounding, BECAs are robust and effective. Overall, there is no reason to believe a universal best BECA exist, and BECAs are compromised in some way or another.

2. Blindly apply the commonly used ComBat strategy leads to poor performance when batch-class design is unbalanced, especially in the severely unbalanced case. Our proposed "CS-ComBat" approach not only readily outperforms the blind approach to ComBat, it also robust in preserving class effects when CEP is high. Besides, rebalancing data with SMOTE first following by different BECAs can also lead to better performance than doing batch effect correction on unbalanced data alone.

3. ComBat results in incomplete removal of batch effects under the small sample size scenario, and our proposed "CS-ComBat" can remove batch effects more thoroughly. Unfortunately, more thorough removal of batch effects results in tradeoffs for downstream statistical feature selection. CS-ComBat is good at recall, but has lower precision than native ComBat. Hence, the decision when to use which should be dictated more by analytical need, than the need to thoroughly eradicate batch effects.

4. Batch effects correction at peptide level or protein level can remove batch effects well and preserve class effects, and also gain good inter-sampling similarity and statistical feature selection results. Both lead to an accurate final protein level.

When to correct batch effects at peptide level or protein level rely on analytical need: if we aim to understand splice variants batch effects correction at protein level is too late, thus, we favor batch effects correction at peptide level. If we aim to study differential protein expression, then correction at either peptide level or protein level can be used.

6.2 Innovations

1. We clarified the advantages and disadvantages of the current mainstream BECAs and their application situations. These findings will help researchers better understand batch effects and choose appropriate methods to reduce the impact of batch effects on their experiments, and help identify and identify disease-related differential genes and drug targets.

2. Aiming at the shortcomings of the existing batch effect correction algorithm (ComBat), we proposed an alternative procedure to ComBat, the class-specific ComBat (CS-ComBat) and demonstrated its applicability in specific situations. Which might provide potential effective batch effects correction methods to this filed.

3. In the proteomics data, the impact of batch effects correction at the peptide level and protein level was explored. Allowing researchers to pay more attention to the batch effects issue and alleviate the impaction of batch effects to their experiment based on their aim.

6.3 Prospects

Even though our results are based on simulated high-throughput data and real high-throughput data, they may not be sufficient to cover all the heterogeneity of the generated data in the high-throughput technology field. Further study will include spiked in high-throughput data (where the true concentration is known in prior), and other real high-throughput data as additional proof.

The use of batch effects correction algorithms (BECAs) and batch effects detection approaches require a certain programming foundation, which is not user-friendly to biologists who are not familiar with programming. Creating interactive and dynamic tools with R/Shiny that integrated these methods (also

included our proposed CS-ComBat) worth further exploration, which could facilitate the extensive use of those methods to non-programmers.

Overcorrection is a common problem that existed to these batch effects correction algorithms (BECAs) that is worthy of future study. For the BECAs, the ability to keep heterogeneity between different classes after correction maybe another direction for further improvement.

Notably, BECAs are not silver bullets, and none of them guarantee to offer an accurate analysis and the correctness of correction results rely on a variety of factors and conditions. In addition, they can not substitute for reasonable experimental design and analysis.

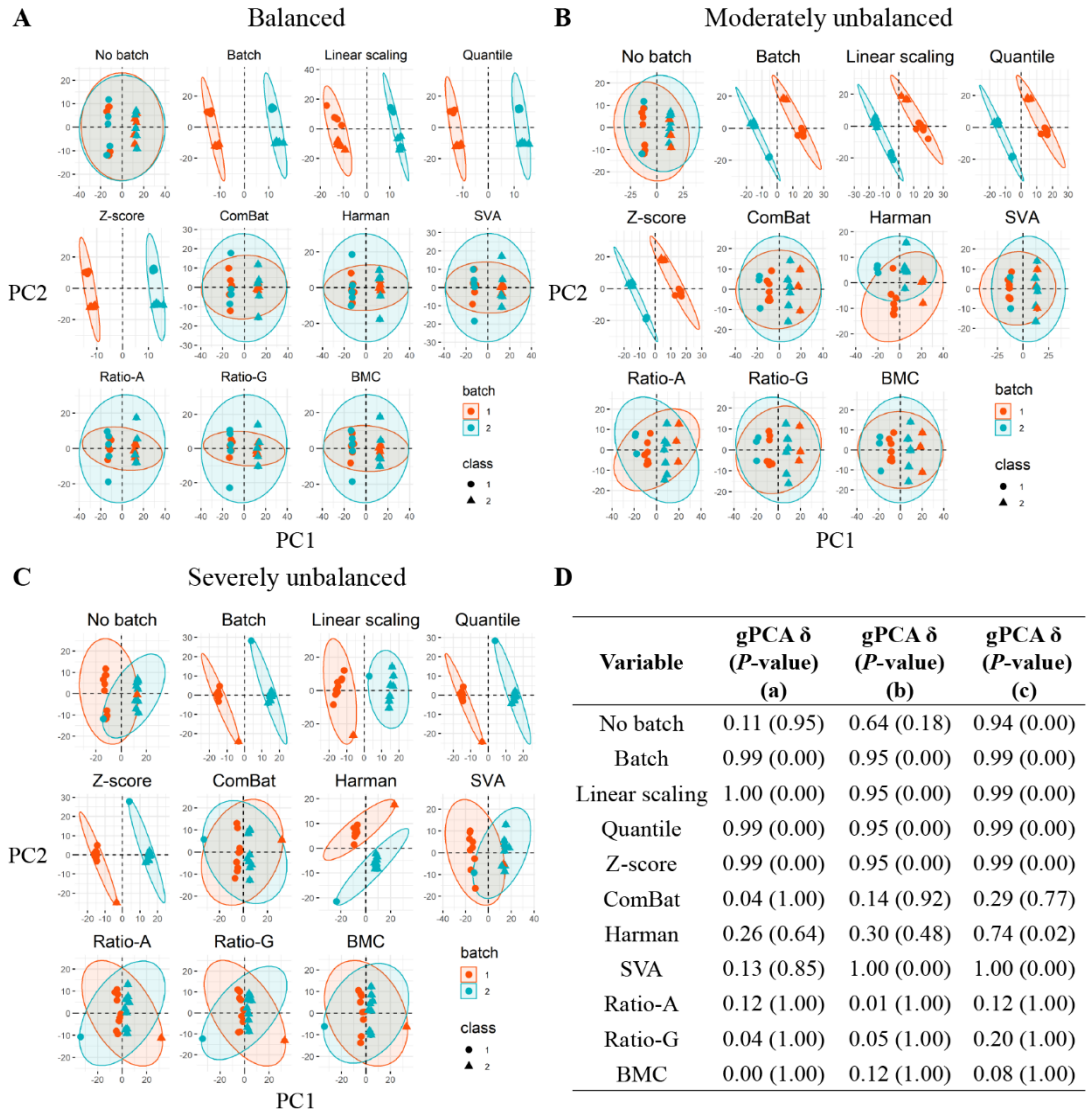
Appendices

Appendix A: Symbol table

Table A-1 Abbreviation and corresponding full name.

Abbreviation	Full Name
AUC	Area under the curve
BECAs	Batch effect-correction algorithms
BMC	Batch mean-centering
BEP	Batch-effect proportion
BERN	Batch effect resistant normalizations
ComBat	Combating batch effects
CS-ComBat	Class specific ComBat
CEP	Class-effect proportion
DWD	distance-weighted discrimination
EB	Empirical bayes
gPCA	Guided-PCA
GFS	Gene fuzzy scoring
PCA	Principal components analysis
PCs	Principal components
pRDA	Partial redundancy analysis
PCT	protein extraction procedures
Ratio-A	Ratio-based approach (arithmetic mean as reference)
Ratio-G	Ratio-based approach (geometric mean as reference)
ROC	Receiver operating characteristic
RUV	Remove unwanted variation
SVA	Surrogate variable analysis
SMOTE	Synthetic minority oversampling technique
SFS	Statistical feature selection
SVD	Singular value decomposition
SWATH-MS	sequential windowed acquisition of all

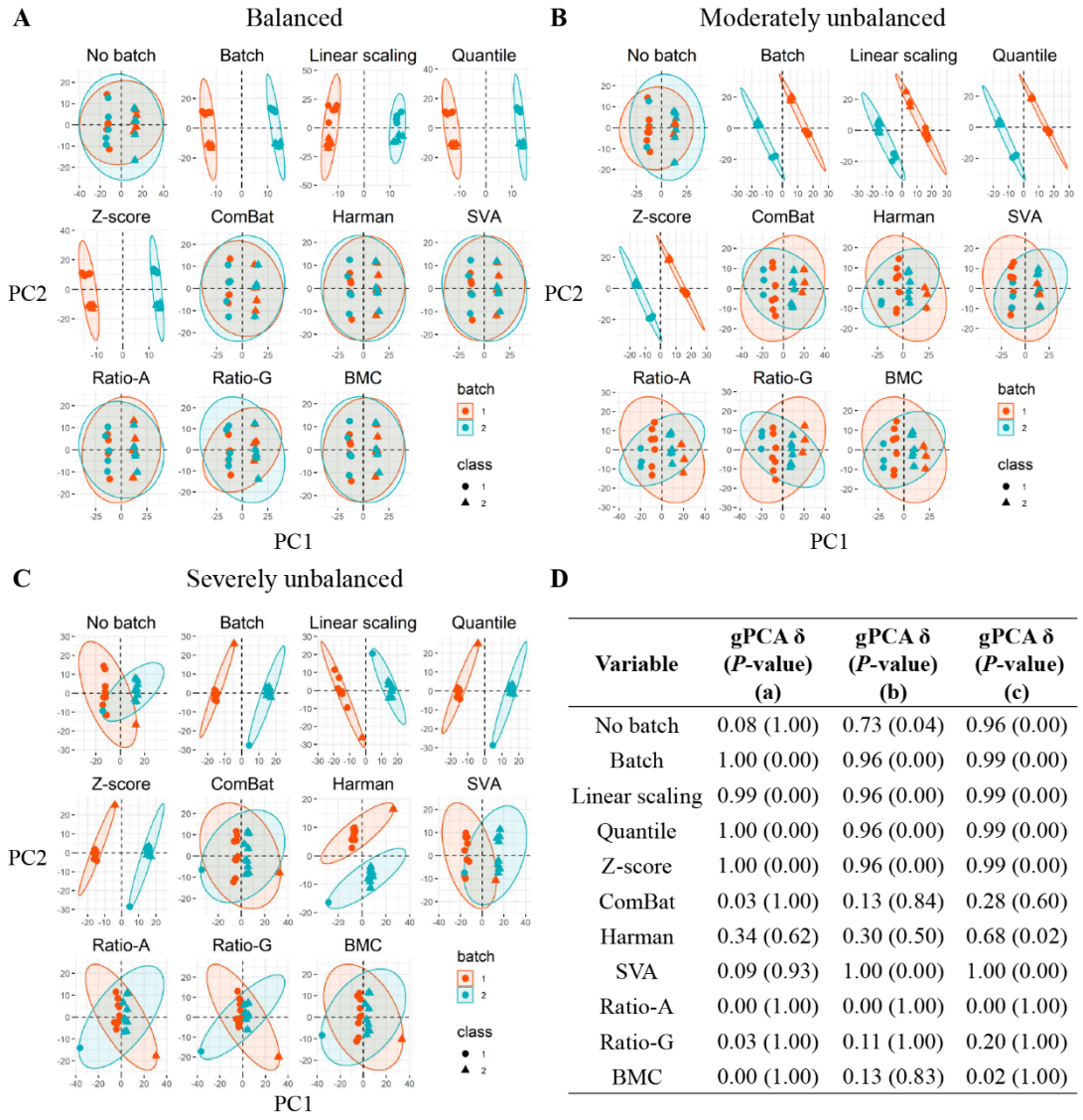
Appendix B: Examining the practical limits of batch effect-correction algorithms: When should you care about batch effects?



D

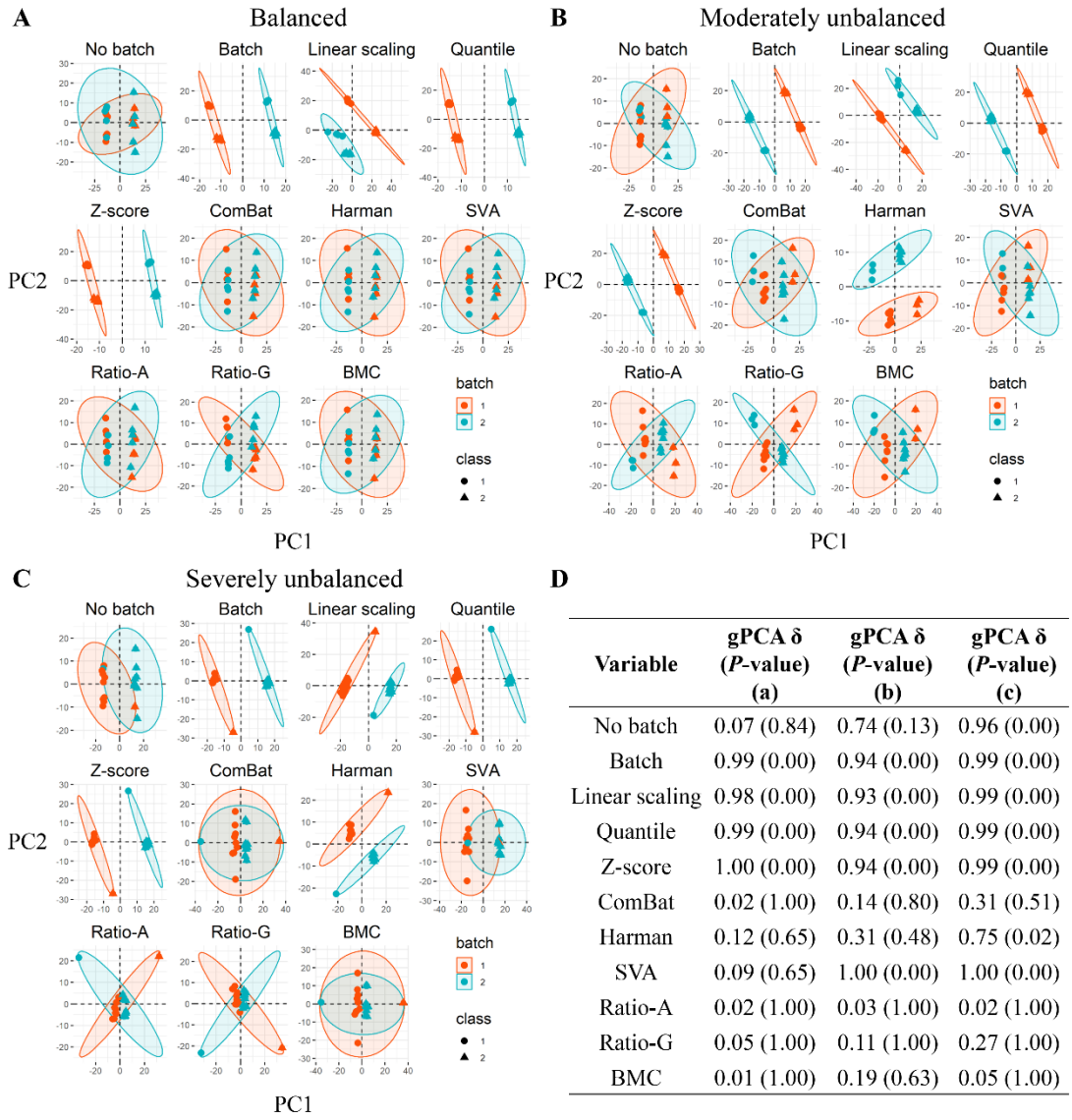
Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.11 (0.95)	0.64 (0.18)	0.94 (0.00)
Batch	0.99 (0.00)	0.95 (0.00)	0.99 (0.00)
Linear scaling	1.00 (0.00)	0.95 (0.00)	0.99 (0.00)
Quantile	0.99 (0.00)	0.95 (0.00)	0.99 (0.00)
Z-score	0.99 (0.00)	0.95 (0.00)	0.99 (0.00)
ComBat	0.04 (1.00)	0.14 (0.92)	0.29 (0.77)
Harman	0.26 (0.64)	0.30 (0.48)	0.74 (0.02)
SVA	0.13 (0.85)	1.00 (0.00)	1.00 (0.00)
Ratio-A	0.12 (1.00)	0.01 (1.00)	0.12 (1.00)
Ratio-G	0.04 (1.00)	0.05 (1.00)	0.20 (1.00)
BMC	0.00 (1.00)	0.12 (1.00)	0.08 (1.00)

Fig. B1. Batch-effect correction in RNA-seq data 2 (The batch-class effects are generated via the non-uniformity assumption). Panels **A-C** shows the 2D-Scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, Z-score) and BECAs (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC). Panel **D** shows the gPCA delta (δ) and associative *P*-values.



Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.08 (1.00)	0.73 (0.04)	0.96 (0.00)
Batch	1.00 (0.00)	0.96 (0.00)	0.99 (0.00)
Linear scaling	0.99 (0.00)	0.96 (0.00)	0.99 (0.00)
Quantile	1.00 (0.00)	0.96 (0.00)	0.99 (0.00)
Z-score	1.00 (0.00)	0.96 (0.00)	0.99 (0.00)
ComBat	0.03 (1.00)	0.13 (0.84)	0.28 (0.60)
Harman	0.34 (0.62)	0.30 (0.50)	0.68 (0.02)
SVA	0.09 (0.93)	1.00 (0.00)	1.00 (0.00)
Ratio-A	0.00 (1.00)	0.00 (1.00)	0.00 (1.00)
Ratio-G	0.03 (1.00)	0.11 (1.00)	0.20 (1.00)
BMC	0.00 (1.00)	0.13 (0.83)	0.02 (1.00)

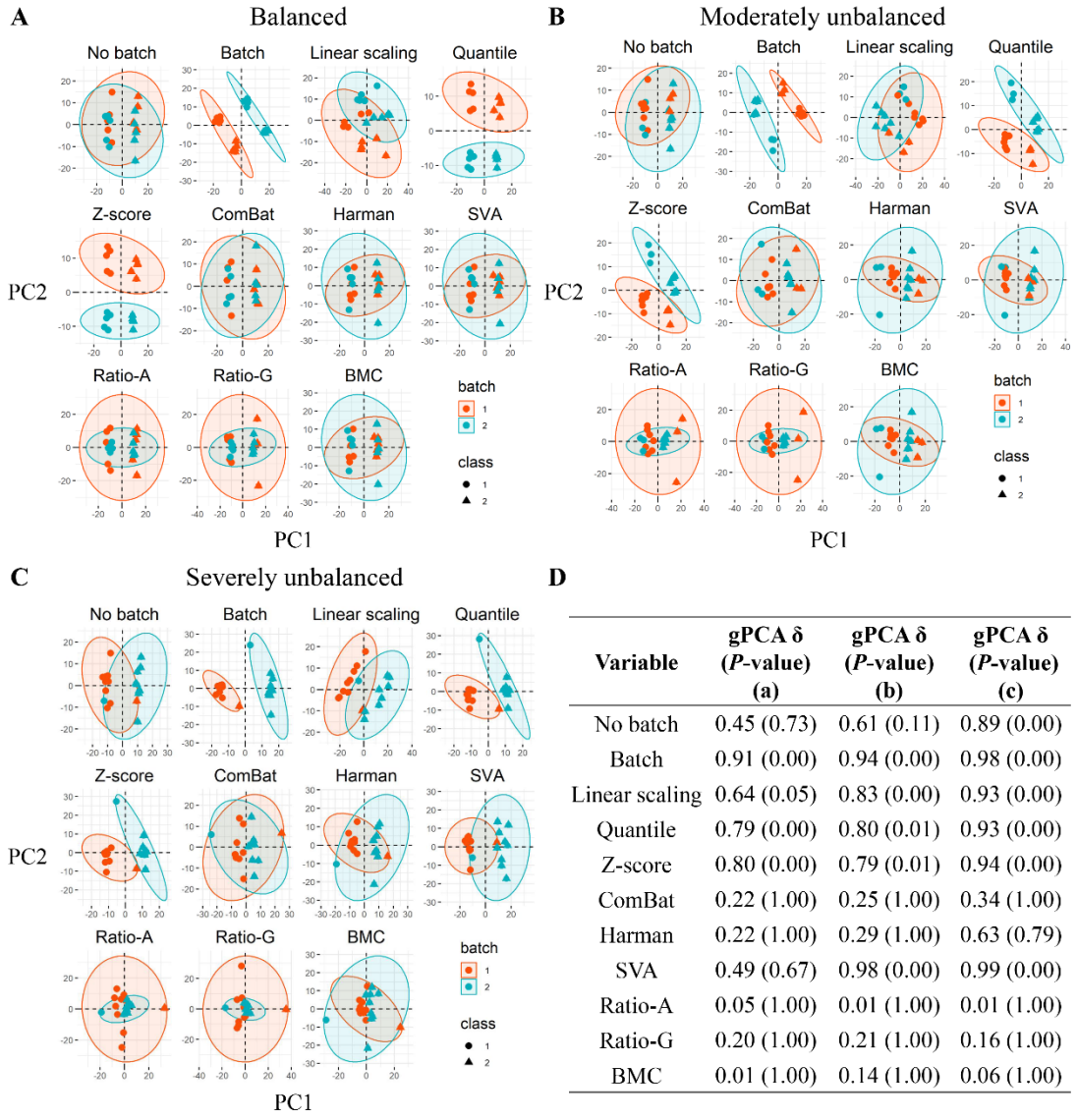
Fig. B2. Batch-effect correction in RNA-seq data 3 (The batch-class effects are generated via the non-uniformity assumption). Panels **A-C** shows the 2D-Scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, Z-score) and BECAs (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC). Panel **D** shows the gPCA delta (δ) and associative *P*-values.



D

Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.07 (0.84)	0.74 (0.13)	0.96 (0.00)
Batch	0.99 (0.00)	0.94 (0.00)	0.99 (0.00)
Linear scaling	0.98 (0.00)	0.93 (0.00)	0.99 (0.00)
Quantile	0.99 (0.00)	0.94 (0.00)	0.99 (0.00)
Z-score	1.00 (0.00)	0.94 (0.00)	0.99 (0.00)
ComBat	0.02 (1.00)	0.14 (0.80)	0.31 (0.51)
Harman	0.12 (0.65)	0.31 (0.48)	0.75 (0.02)
SVA	0.09 (0.65)	1.00 (0.00)	1.00 (0.00)
Ratio-A	0.02 (1.00)	0.03 (1.00)	0.02 (1.00)
Ratio-G	0.05 (1.00)	0.11 (1.00)	0.27 (1.00)
BMC	0.01 (1.00)	0.19 (0.63)	0.05 (1.00)

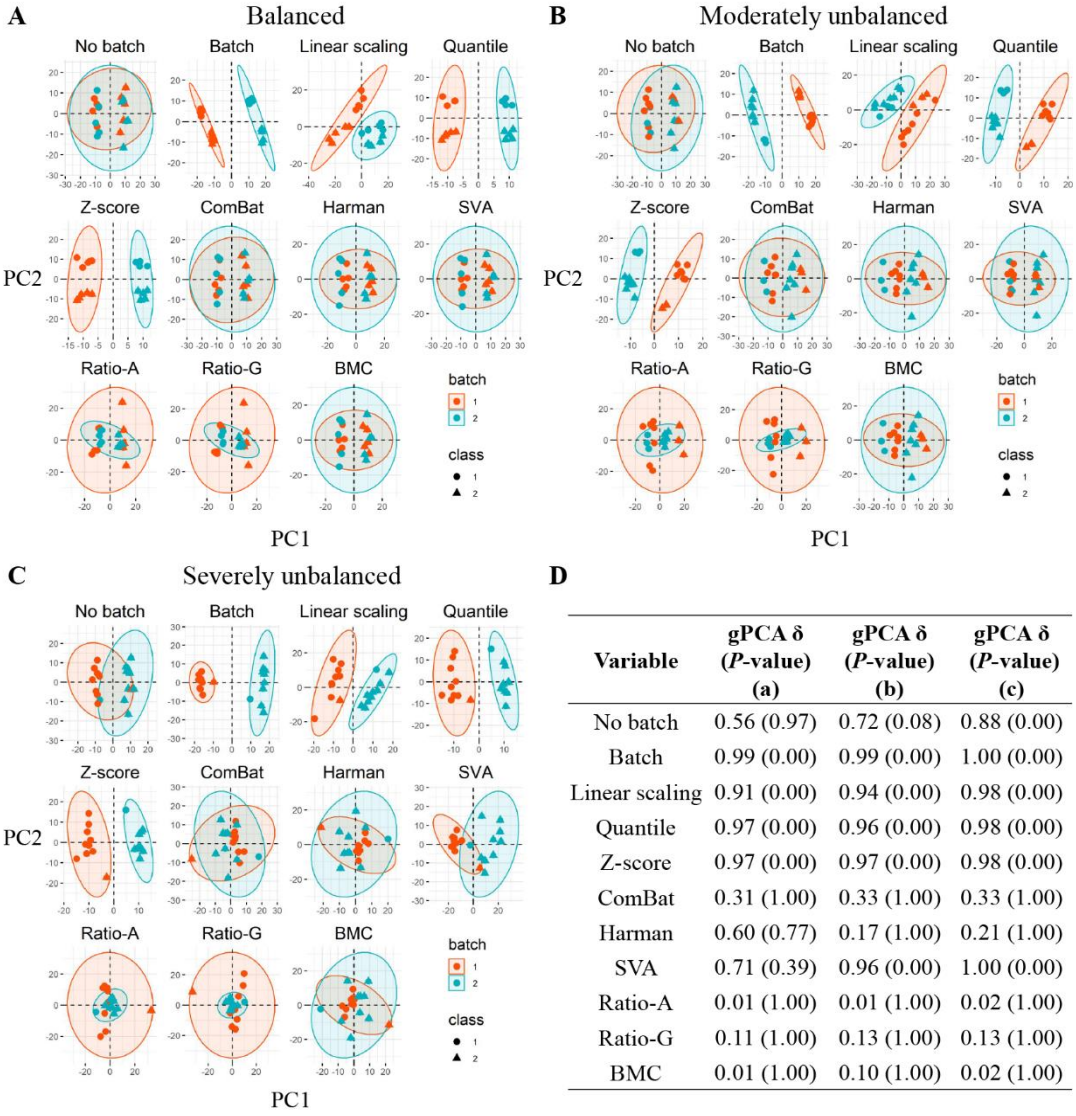
Fig. B3. Batch-effect correction in proteomics data (The batch-class effects are generated via the non-uniformity assumption). Panels **A-C** shows the 2D-Scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, Z-score) and BECAs (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC). Panel **D** shows the gPCA delta (δ) and associative *P*-values.



D

Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.45 (0.73)	0.61 (0.11)	0.89 (0.00)
Batch	0.91 (0.00)	0.94 (0.00)	0.98 (0.00)
Linear scaling	0.64 (0.05)	0.83 (0.00)	0.93 (0.00)
Quantile	0.79 (0.00)	0.80 (0.01)	0.93 (0.00)
Z-score	0.80 (0.00)	0.79 (0.01)	0.94 (0.00)
ComBat	0.22 (1.00)	0.25 (1.00)	0.34 (1.00)
Harman	0.22 (1.00)	0.29 (1.00)	0.63 (0.79)
SVA	0.49 (0.67)	0.98 (0.00)	0.99 (0.00)
Ratio-A	0.05 (1.00)	0.01 (1.00)	0.01 (1.00)
Ratio-G	0.20 (1.00)	0.21 (1.00)	0.16 (1.00)
BMC	0.01 (1.00)	0.14 (1.00)	0.06 (1.00)

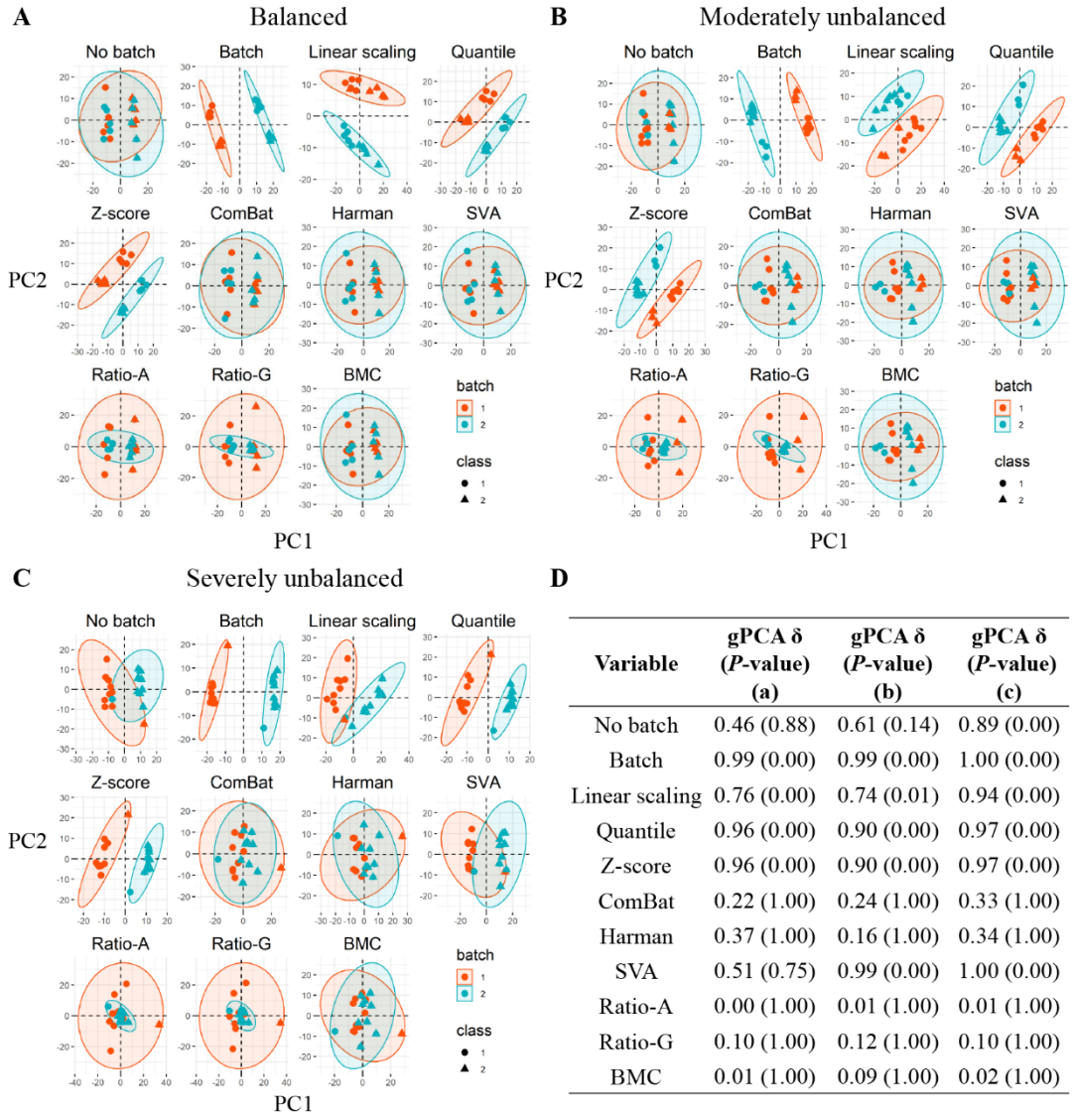
Fig. B4. Batch-effect correction in RNA-seq data 1 (using the Sarah-Langley approach with assumption of uniformity). Panels **A-C** shows the 2D-Scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, Z-score) and BECAs (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC). Panel **D** shows the gPCA delta (δ) and associative *P*-values.



D

Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.56 (0.97)	0.72 (0.08)	0.88 (0.00)
Batch	0.99 (0.00)	0.99 (0.00)	1.00 (0.00)
Linear scaling	0.91 (0.00)	0.94 (0.00)	0.98 (0.00)
Quantile	0.97 (0.00)	0.96 (0.00)	0.98 (0.00)
Z-score	0.97 (0.00)	0.97 (0.00)	0.98 (0.00)
ComBat	0.31 (1.00)	0.33 (1.00)	0.33 (1.00)
Harman	0.60 (0.77)	0.17 (1.00)	0.21 (1.00)
SVA	0.71 (0.39)	0.96 (0.00)	1.00 (0.00)
Ratio-A	0.01 (1.00)	0.01 (1.00)	0.02 (1.00)
Ratio-G	0.11 (1.00)	0.13 (1.00)	0.13 (1.00)
BMC	0.01 (1.00)	0.10 (1.00)	0.02 (1.00)

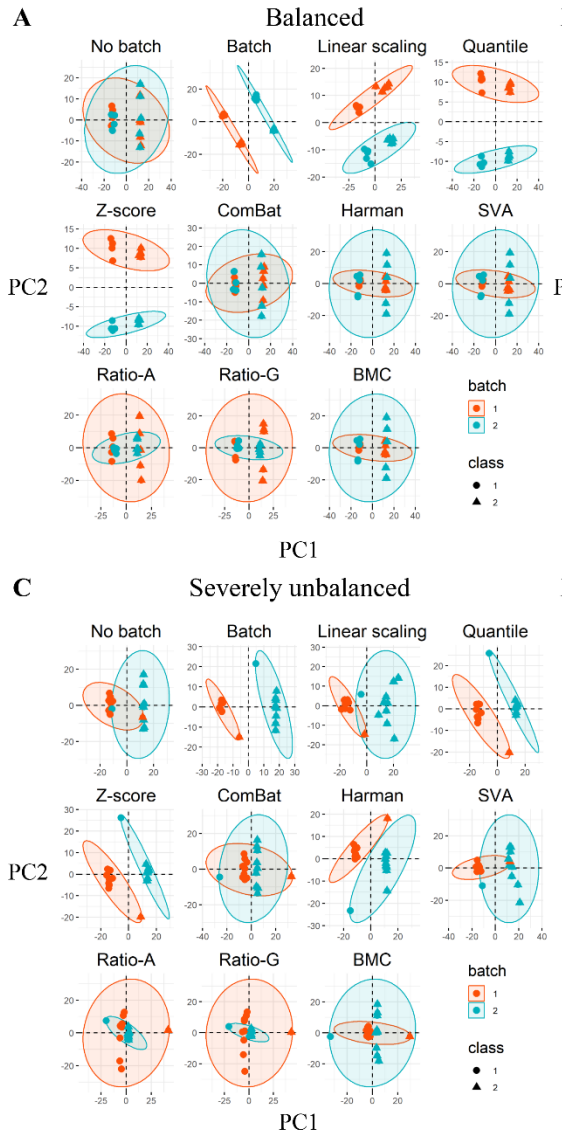
Fig. B5. Batch-effect correction in RNA-seq data 2 (using the Sarah-Langley approach with assumption of uniformity). Panels **A-C** shows the 2D-Scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, Z-score) and BECAs (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC). Panel **D** shows the gPCA delta (δ) and associative *P*-values.



D

Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.46 (0.88)	0.61 (0.14)	0.89 (0.00)
Batch	0.99 (0.00)	0.99 (0.00)	1.00 (0.00)
Linear scaling	0.76 (0.00)	0.74 (0.01)	0.94 (0.00)
Quantile	0.96 (0.00)	0.90 (0.00)	0.97 (0.00)
Z-score	0.96 (0.00)	0.90 (0.00)	0.97 (0.00)
ComBat	0.22 (1.00)	0.24 (1.00)	0.33 (1.00)
Harman	0.37 (1.00)	0.16 (1.00)	0.34 (1.00)
SVA	0.51 (0.75)	0.99 (0.00)	1.00 (0.00)
Ratio-A	0.00 (1.00)	0.01 (1.00)	0.01 (1.00)
Ratio-G	0.10 (1.00)	0.12 (1.00)	0.10 (1.00)
BMC	0.01 (1.00)	0.09 (1.00)	0.02 (1.00)

Fig. B6. Batch-effect correction in RNA-seq data 3 (using the Sarah-Langley approach with assumption of uniformity). Panels **A-C** shows the 2D-Scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, Z-score) and BECAs (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC). Panel **D** shows the gPCA delta (δ) and associative *P*-values.



Variable	gPCA δ (*P*-value) (a)	gPCA δ (*P*-value) (b)	gPCA δ (*P*-value) (c)
No batch	0.25 (0.78)	0.61 (0.03)	0.92 (0.00)
Batch	0.86 (0.00)	0.94 (0.00)	0.99 (0.00)
Linear scaling	0.52 (0.14)	0.81 (0.01)	0.97 (0.00)
Quantile	0.62 (0.05)	0.76 (0.01)	0.94 (0.00)
Z-score	0.61 (0.06)	0.76 (0.00)	0.94 (0.00)
ComBat	0.13 (1.00)	0.10 (1.00)	0.19 (1.00)
Harman	0.05 (1.00)	0.29 (0.99)	0.77 (0.01)
SVA	0.25 (1.00)	0.99 (0.00)	1.00 (0.00)
Ratio-A	0.01 (1.00)	0.03 (1.00)	0.01 (1.00)
Ratio-G	0.07 (1.00)	0.16 (1.00)	0.11 (1.00)
BMC	0.02 (1.00)	0.27 (1.00)	0.10 (1.00)

Fig. B7. Batch-effect correction in proteomics data (using the Sarah-Langley approach with assumption of uniformity). Panels **A-C** shows the 2D-Scatterplots for the simulated data (no batch), incorporation of batch effects (batch), and batch correction performance in conventional normalization (linear scaling, quantile, Z-score) and BECAs (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC). Panel **D** shows the gPCA delta (δ) and associative *P*-values.

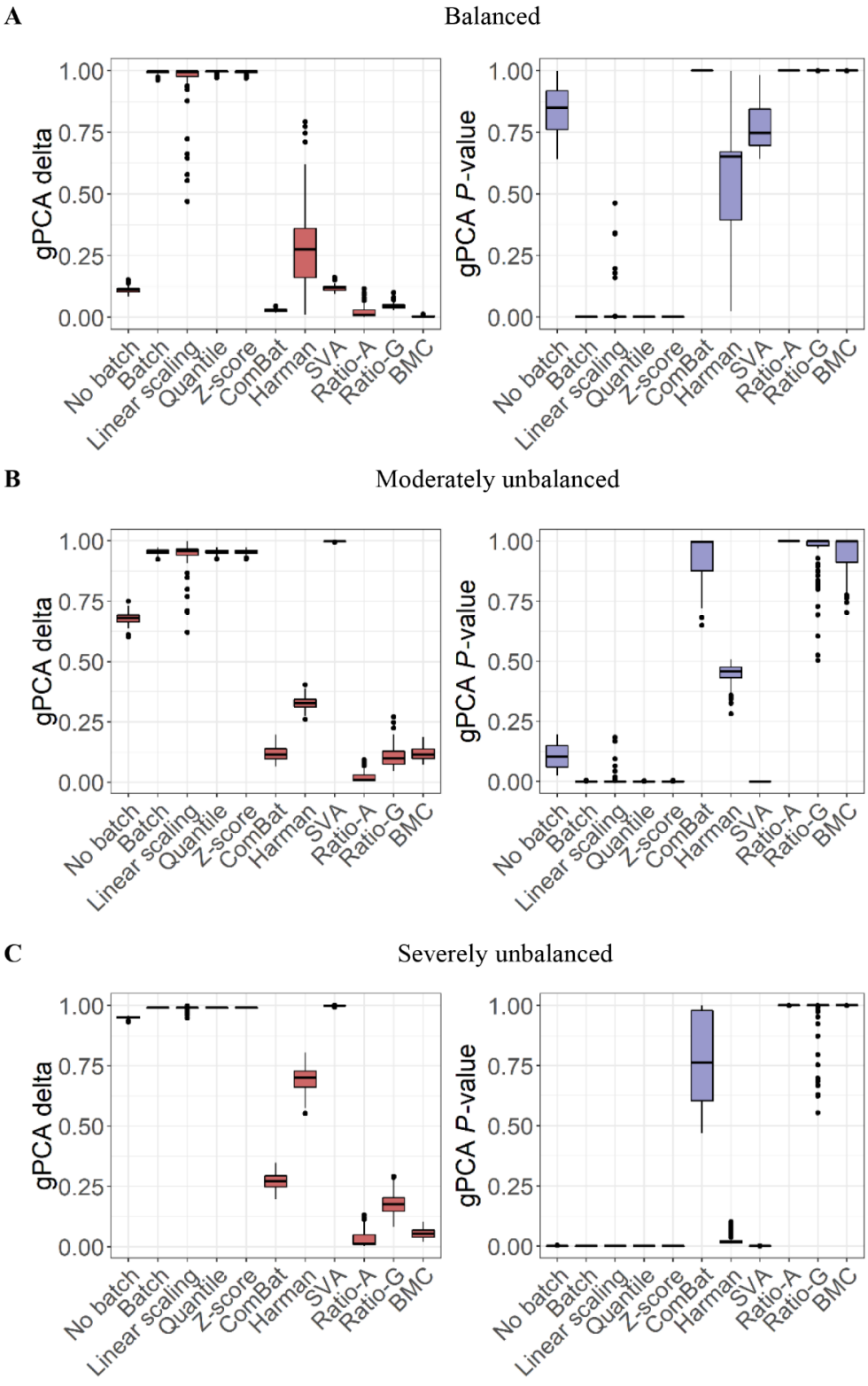


Fig. B8. gPCA delta and *P*-value distribution in batch-effect correction in RNA-seq data 2 (100 simulations). The batch-class effects are generated via the non-uniformity assumption.

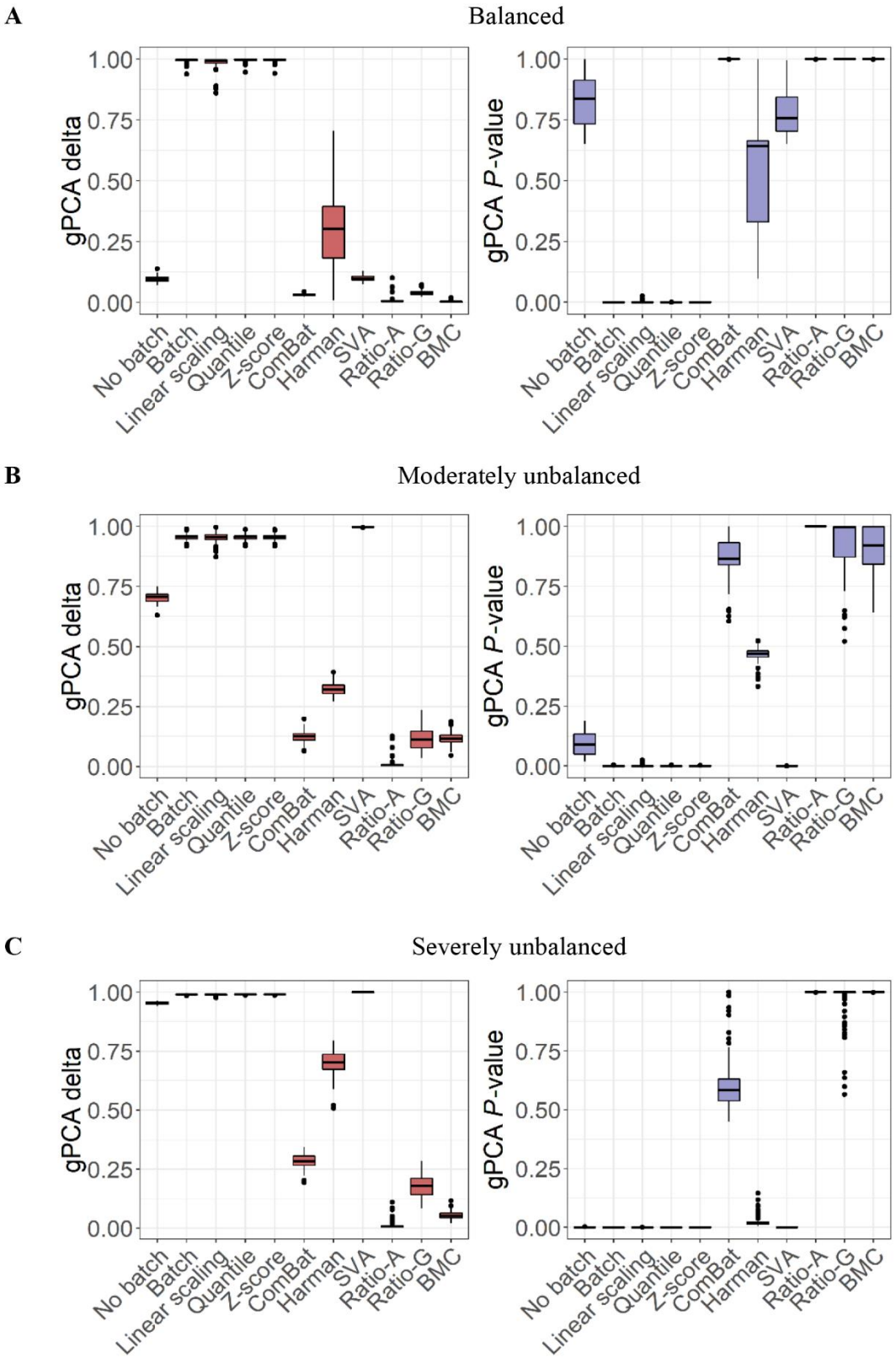


Fig. B9. gPCA delta and *P*-value distribution in batch-effect correction in RNA-seq data 3 (100 simulations). The batch-class effects are generated via the non-uniformity assumption.

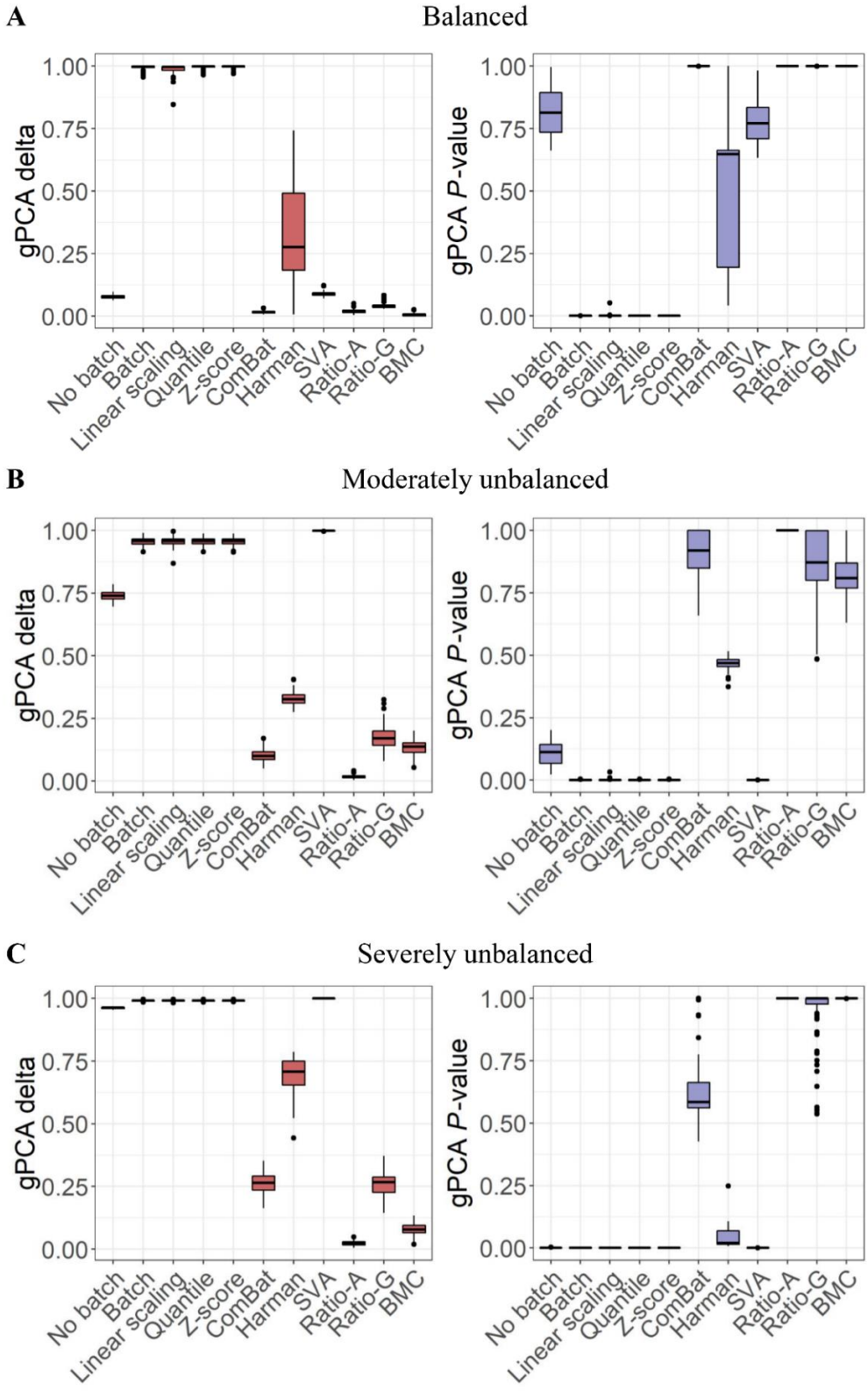


Fig. B10. gPCA delta and *P*-value distribution in batch-effect correction in proteomics data (100 simulations). The batch-class effects are generated via the non-uniformity assumption.

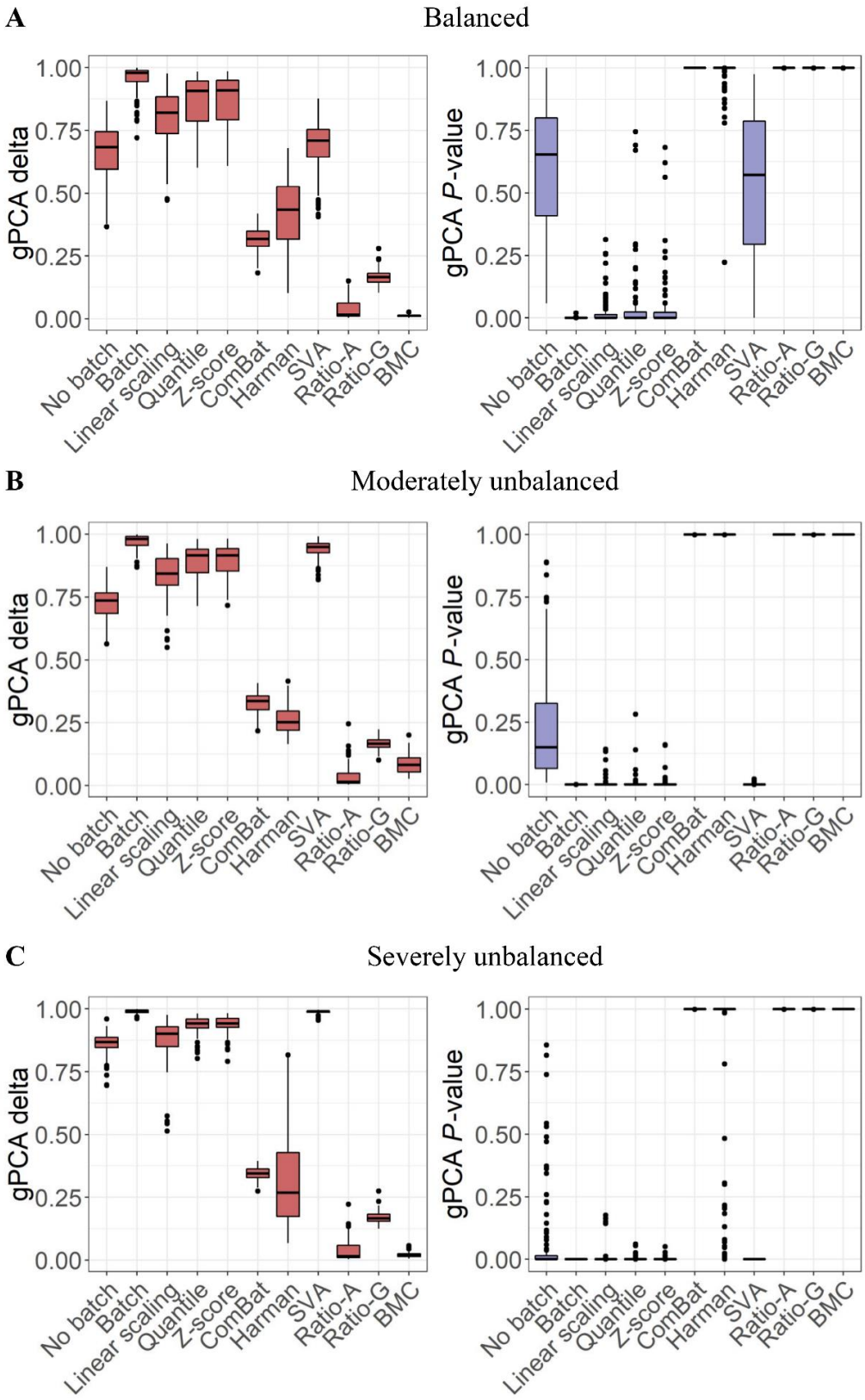


Fig. B11. gPCA delta and *P*-value distribution in batch-effect correction in RNA-seq data (100 simulations). The batch-class effects are generated via the uniformity assumption.

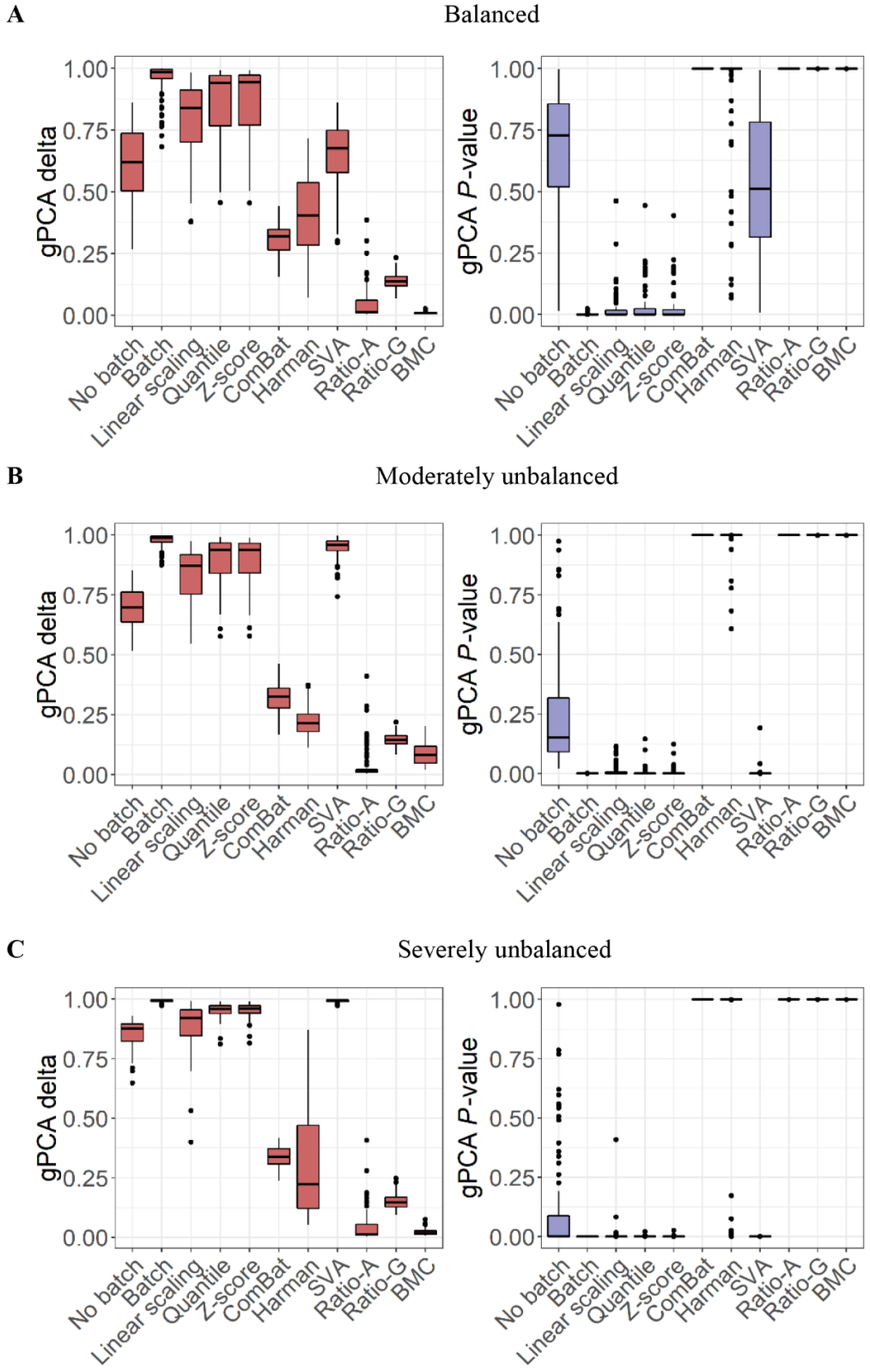


Fig. B12. gPCA delta and *P*-value distribution in batch-effect correction in RNA-seq data 2 (100 simulations). The batch-class effects are generated via the uniformity assumption.

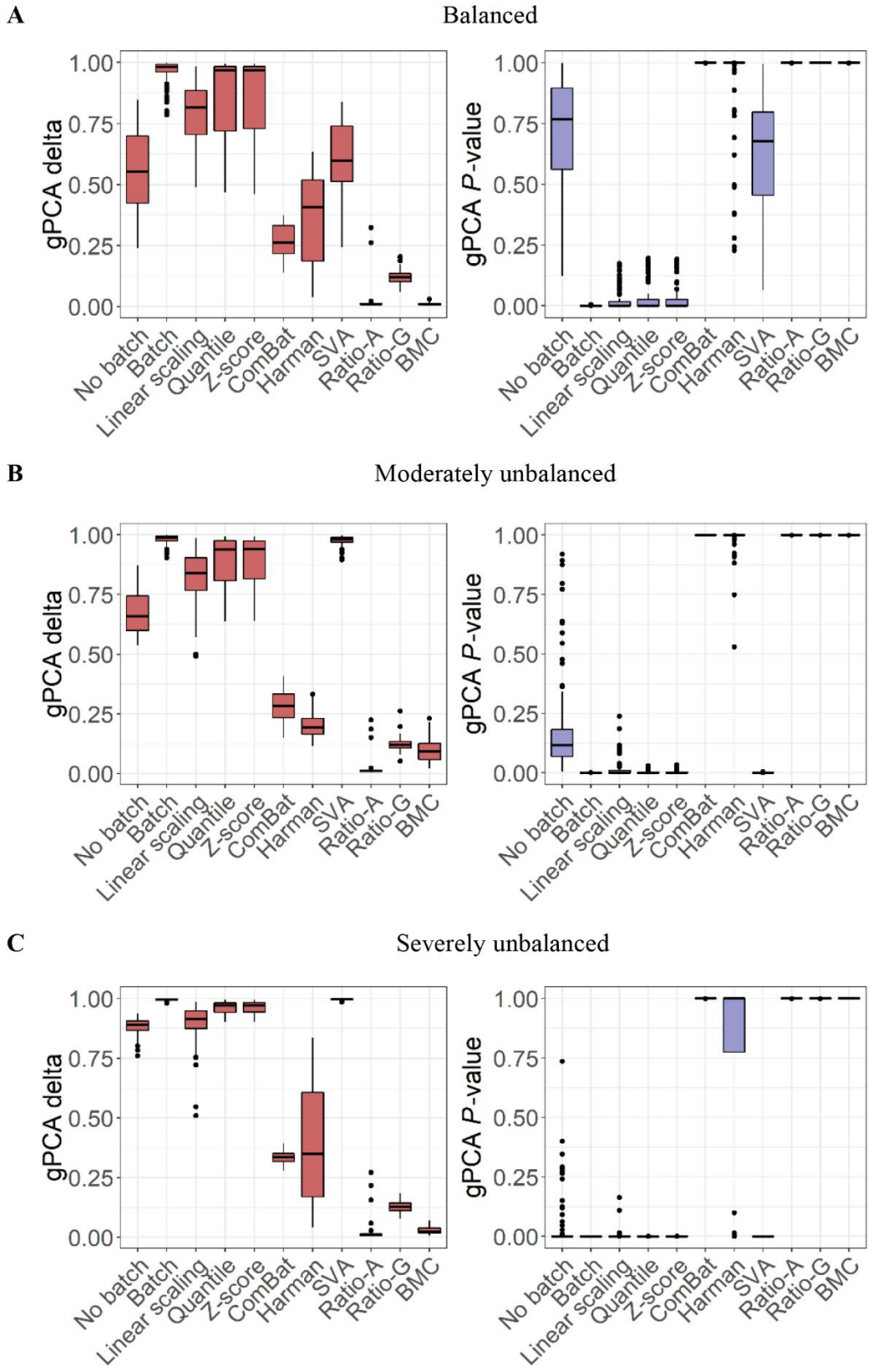


Fig. B13. gPCA delta and *P*-value distribution in batch-effect correction in RNA-seq data 3 (100 simulations). The batch-class effects are generated via the uniformity assumption.

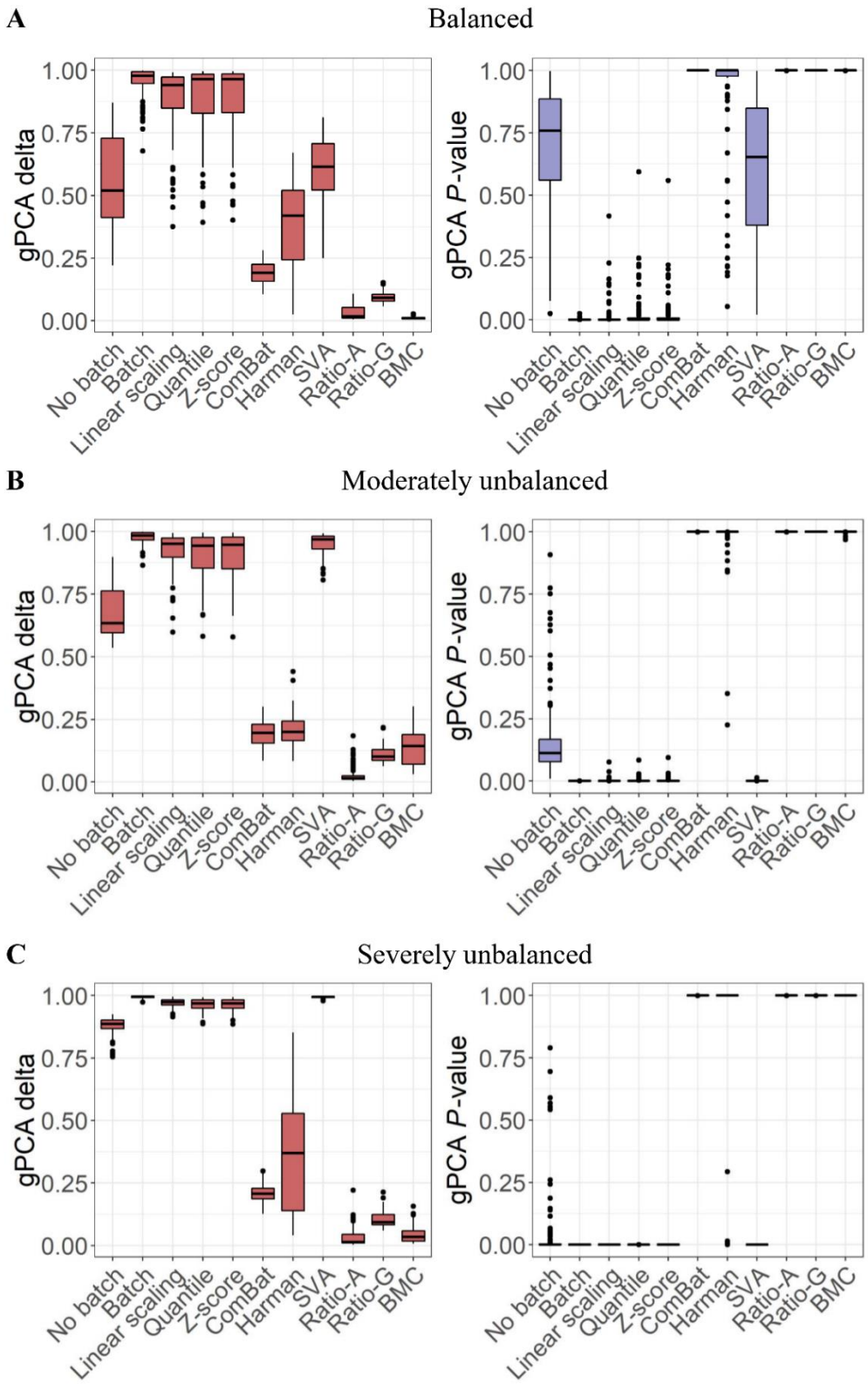


Fig. B14. gPCA delta and *P*-value distribution in batch-effect correction in proteomics data (100 simulations). The batch-class effects are generated via the uniformity assumption.

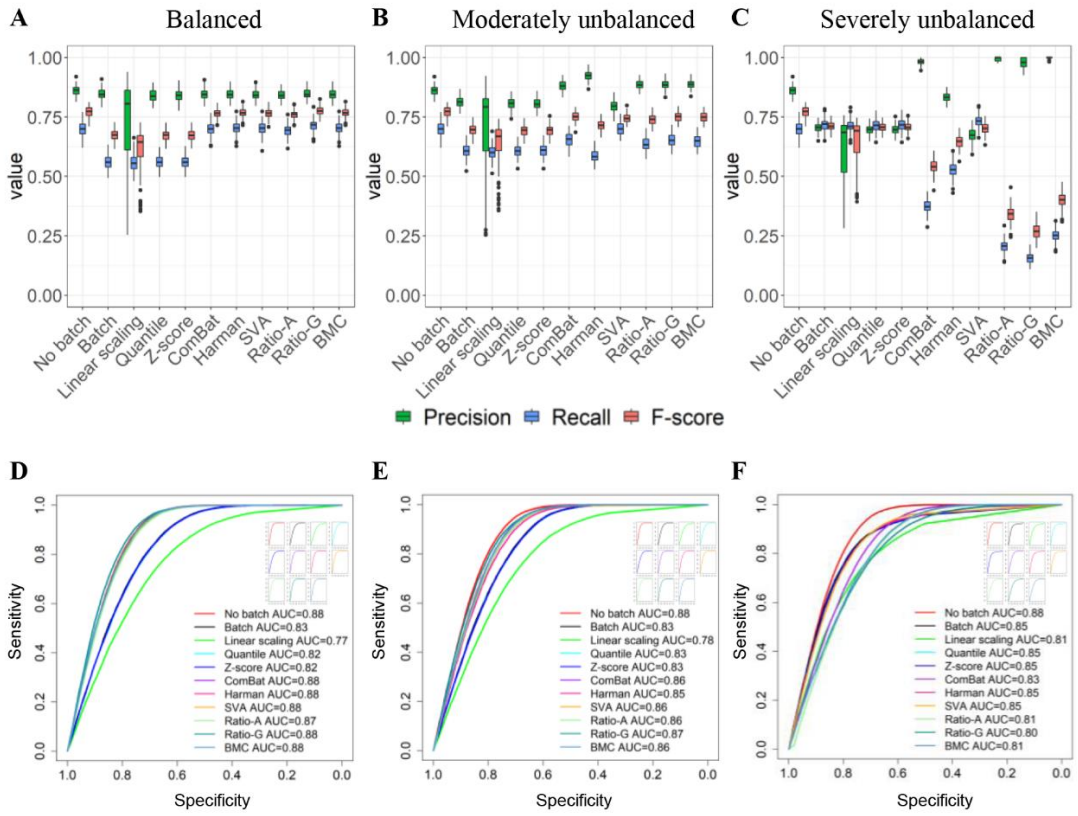


Fig. B15. Performance evaluation on precision/recall and ROC (RNA-seq data 2, Non-uniformity assumption for batch-class, 100 simulations). Panels **A** to **C** shows the precision, recall and F-score distributions. Panels **D** to **F** shows the average ROC curves and the corresponding average AUC values.

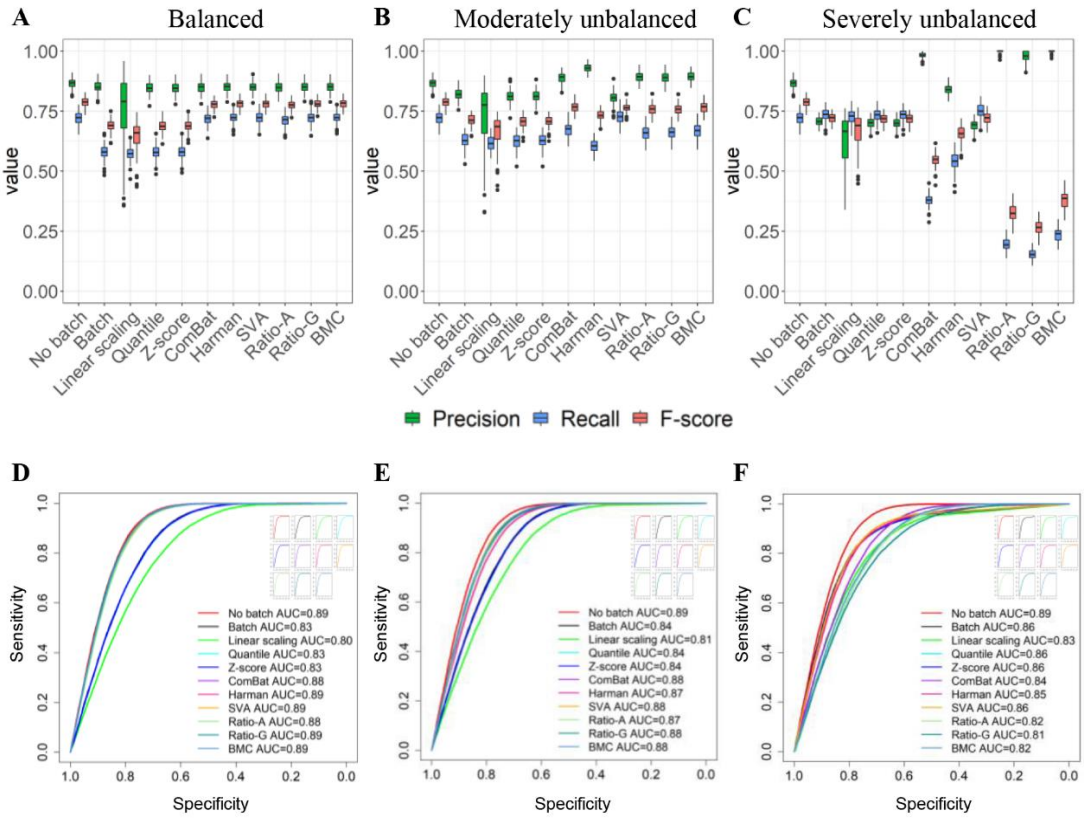


Fig. B16. Performance evaluation on precision/recall and ROC (RNA-seq data 3, Non-uniformity assumption for batch-class, 100 simulations). Panels **A** to **C** shows the precision, recall and F-score distributions. Panels **D** to **F** shows the average ROC curves and the corresponding average AUC values.

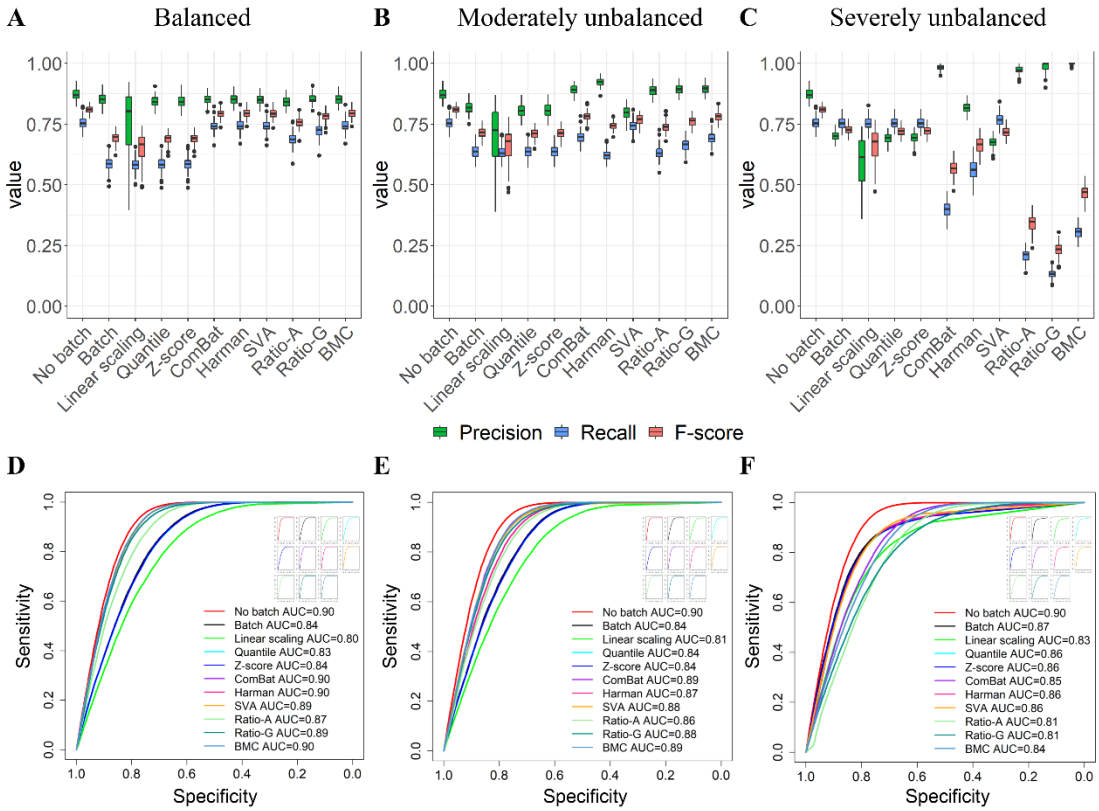


Fig. B17. Performance evaluation on precision/recall and ROC (Proteomics, Non-uniformity assumption for batch-class, 100 simulations). Panels **A** to **C** shows the precision, recall and F-score distributions. Panels **D** to **F** shows the average ROC curves and the corresponding average AUC values.

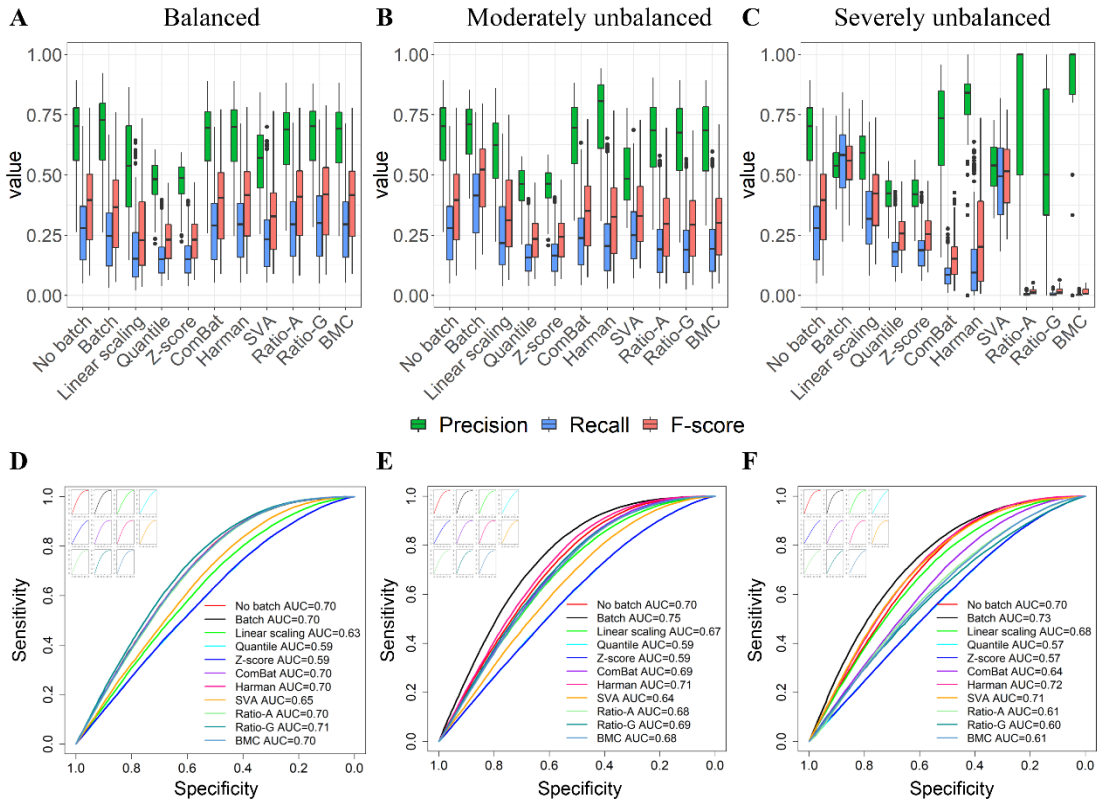


Fig. B18. Performance evaluation on precision/recall and ROC (RNA-seq data 1, Uniformity assumption for batch-class, 100 simulations). Panels **A** to **C** shows the precision, recall and F-score distributions. Panels **D** to **F** shows the average ROC curves and the corresponding average AUC values.

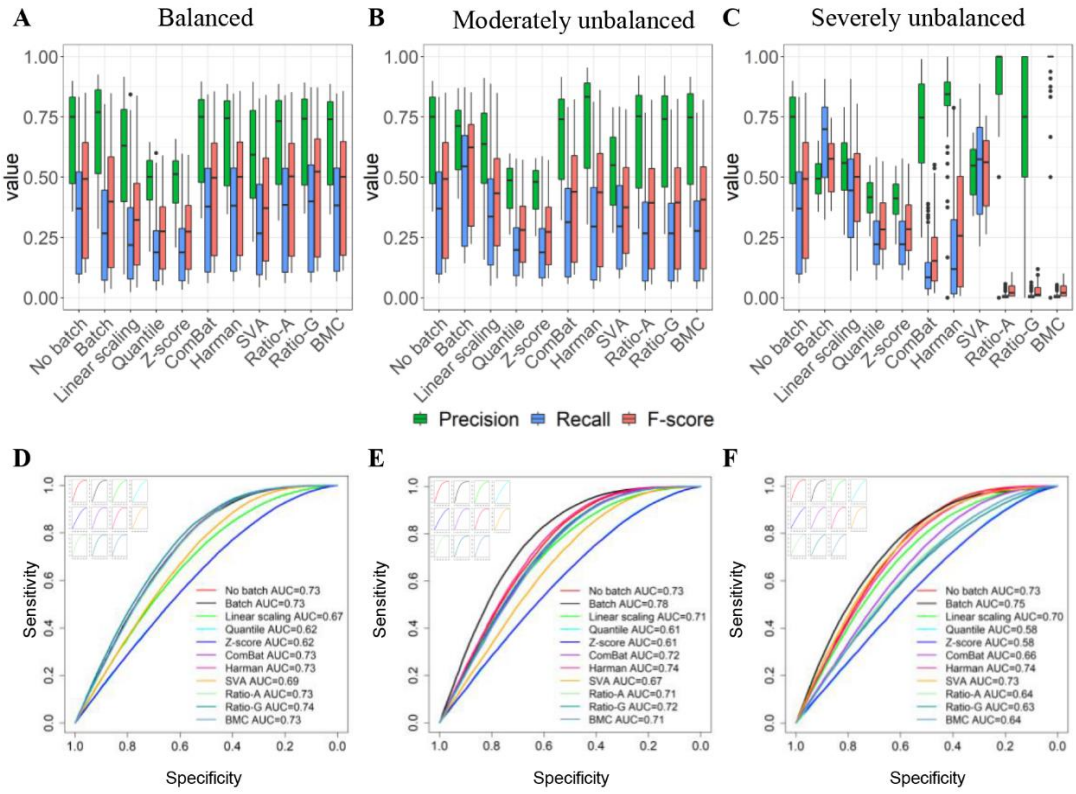


Fig. B19. Performance evaluation on precision/recall and ROC (RNA-seq data 2, Uniformity assumption for batch-class, 100 simulations). Panels **A** to **C** shows the precision, recall and F-score distributions. Panels **D** to **F** shows the average ROC curves and the corresponding average AUC values.

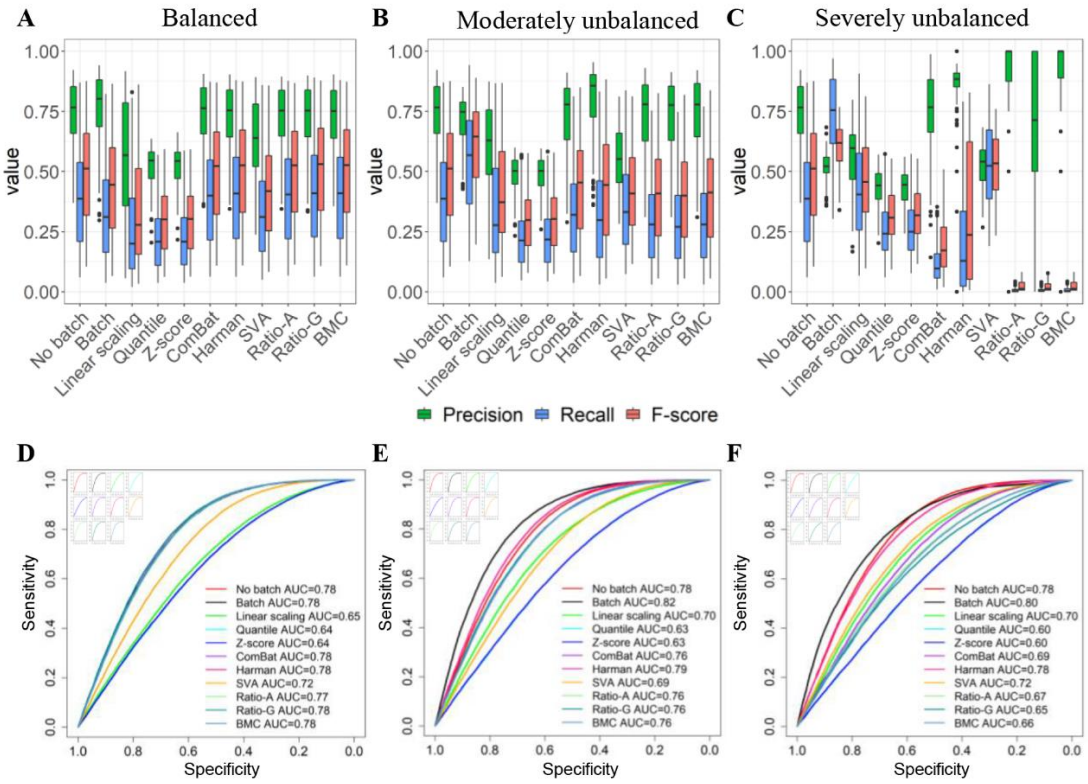


Fig. B20. Performance evaluation on precision/recall and ROC (RNA-seq data 3, Uniformity assumption for batch-class, 100 simulations). Panels **A** to **C** shows the precision, recall and F-score distributions. Panels **D** to **F** shows the average ROC curves and the corresponding average AUC values.

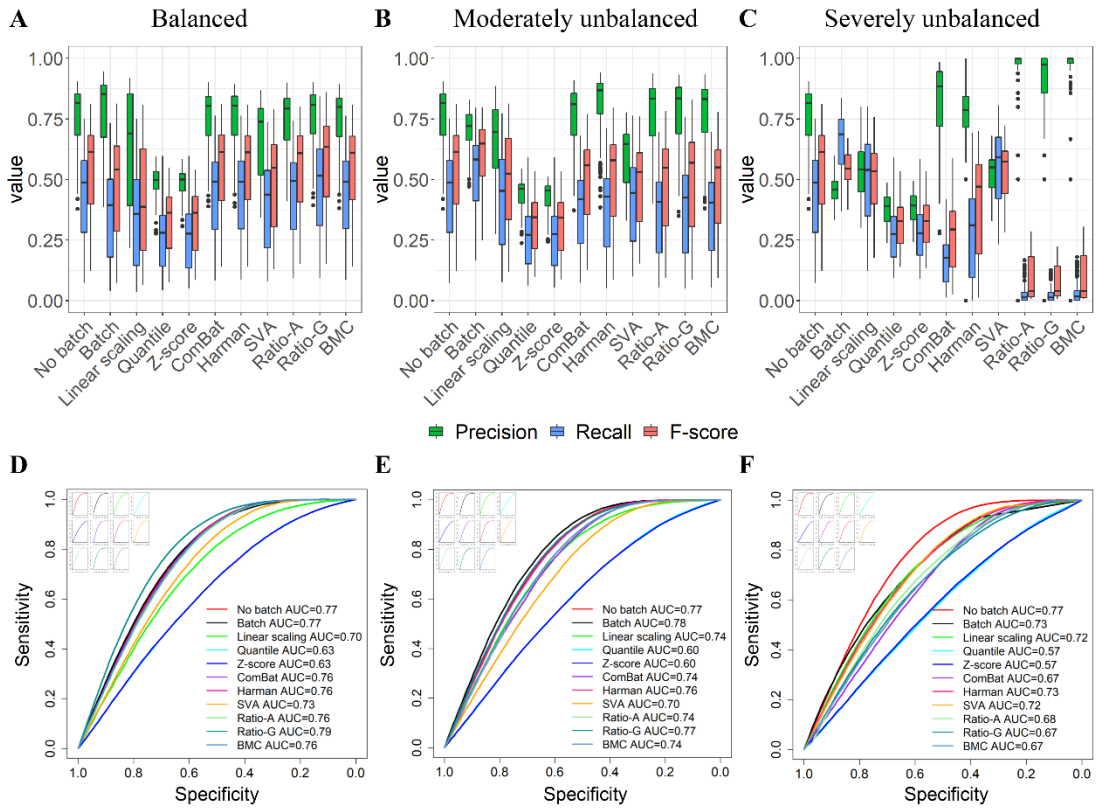


Fig. B21. Performance evaluation on precision/recall and ROC (Proteomics, Uniformity assumption for batch-class, 100 simulations). Panels **A** to **C** shows the precision, recall and F-score distributions. Panels **D** to **F** shows the average ROC curves and the corresponding average AUC values.

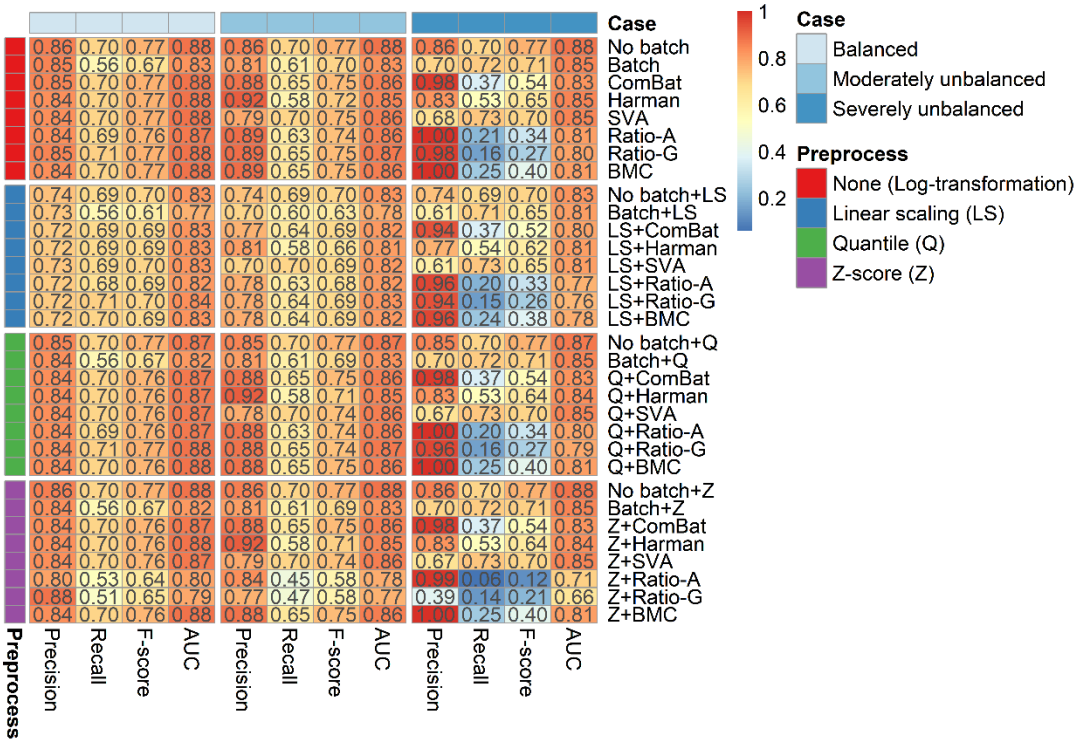


Fig. B22. The impact of prior normalization on BECAs (RNA-seq data 2, Non-uniformity assumption for batch-class, 100 simulations).

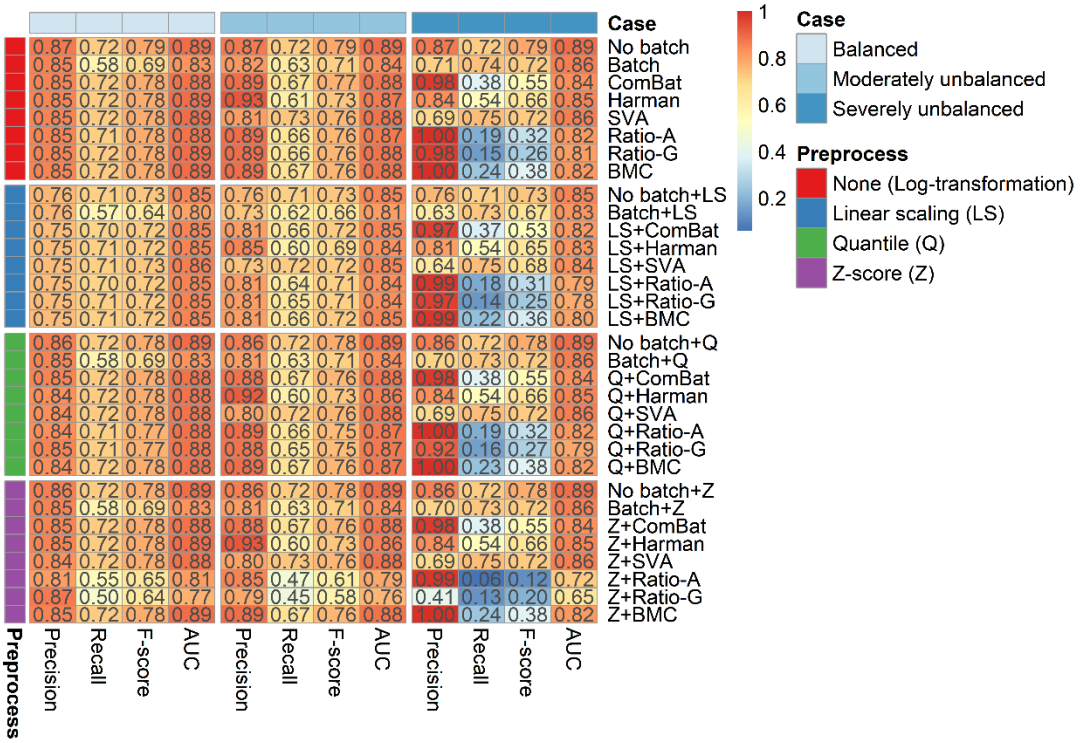


Fig. B23. The impact of prior normalization on BECAs (RNA-seq data 3, Non-uniformity assumption for batch-class, 100 simulations).

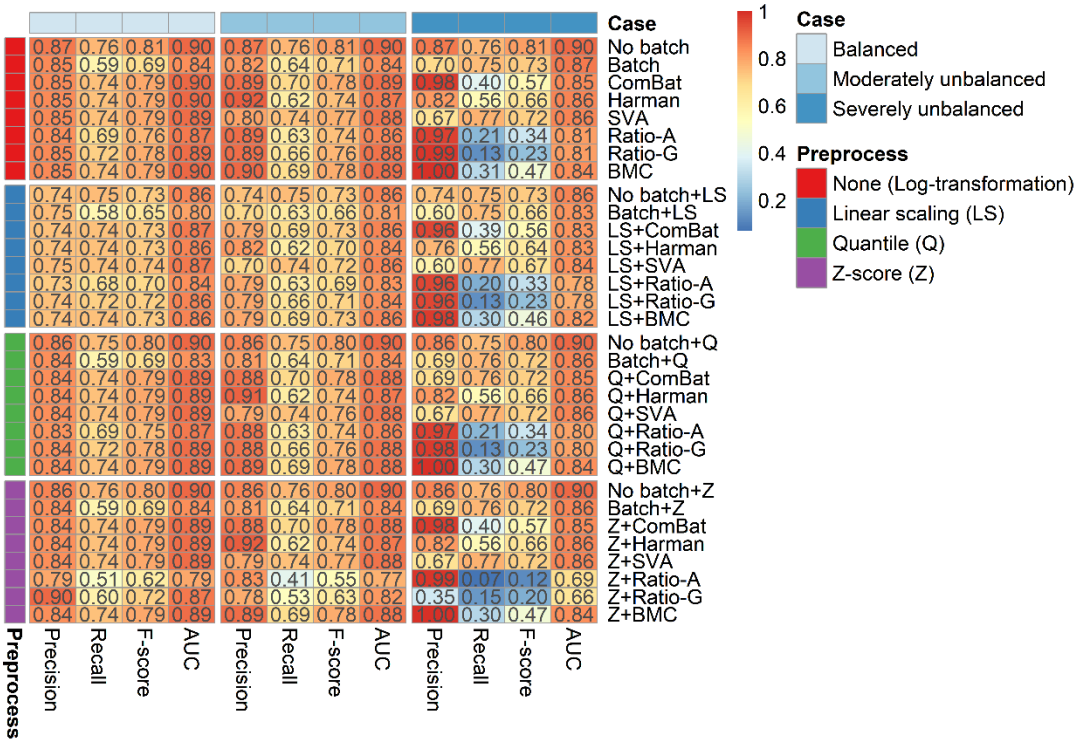


Fig. B24. The impact of prior normalization on BECAs (Proteomics, Non-uniformity assumption for batch-class, 100 simulations).

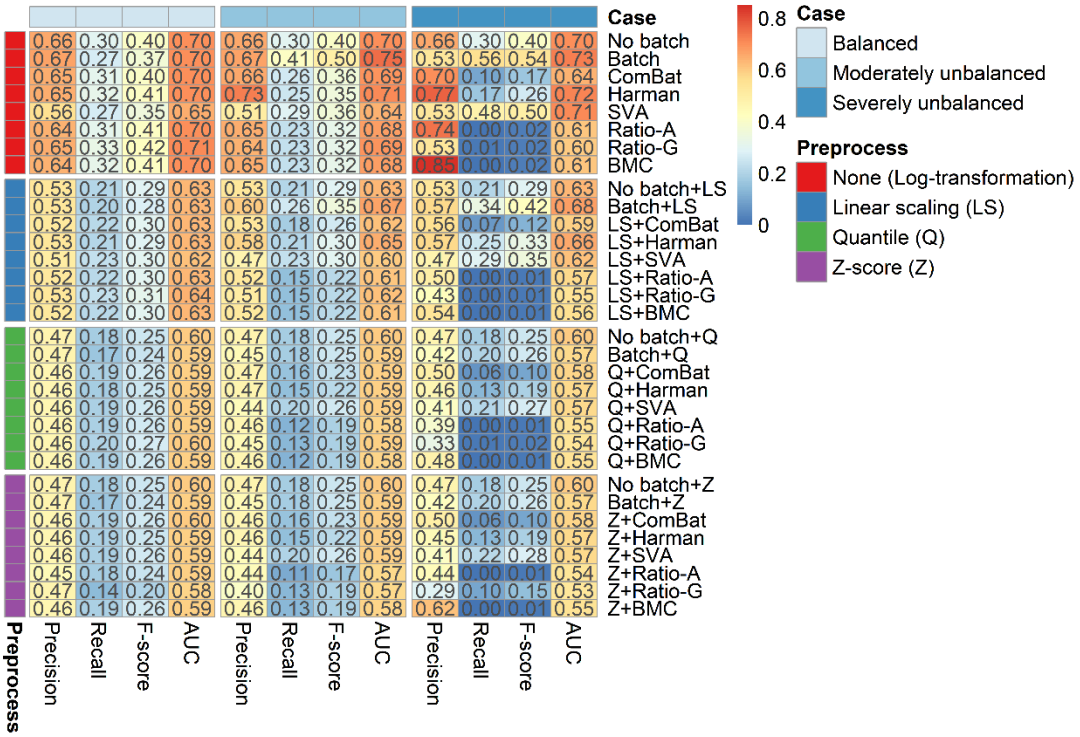


Fig. B25. The impact of prior normalization on BECAs (RNA-seq data 1, Uniformity assumption for batch-class, 100 simulations)**.**

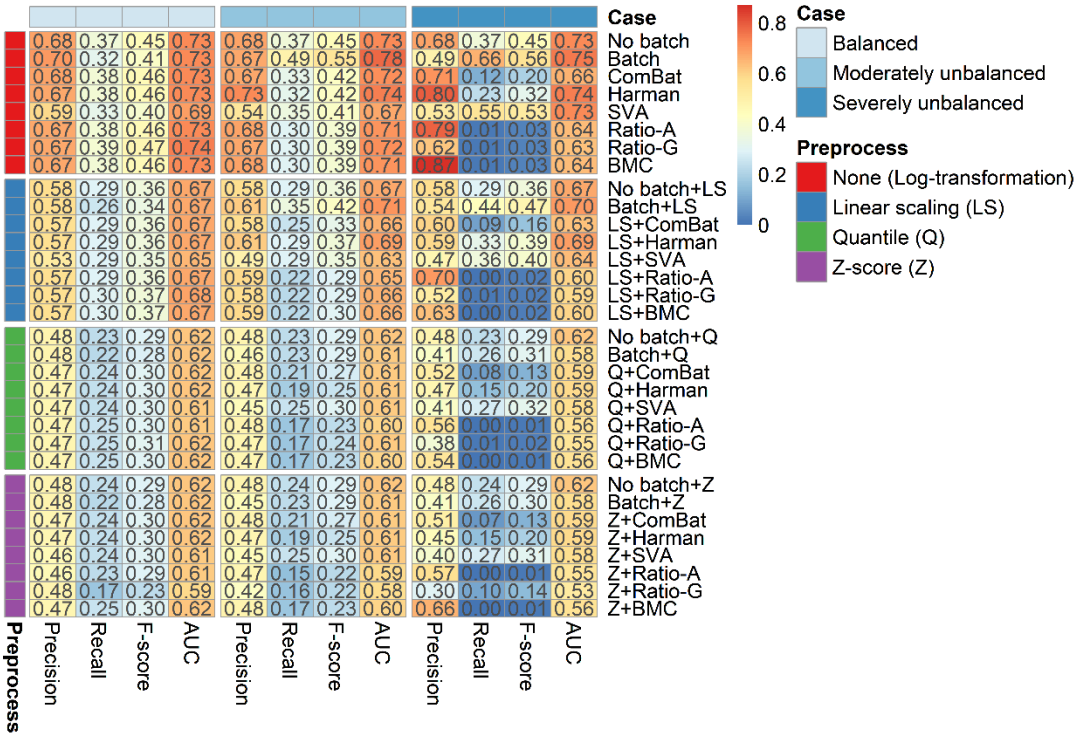


Fig. B26. The impact of prior normalization on BECAs (RNA-seq data 2, Uniformity assumption for batch-class, 100 simulations)**.**

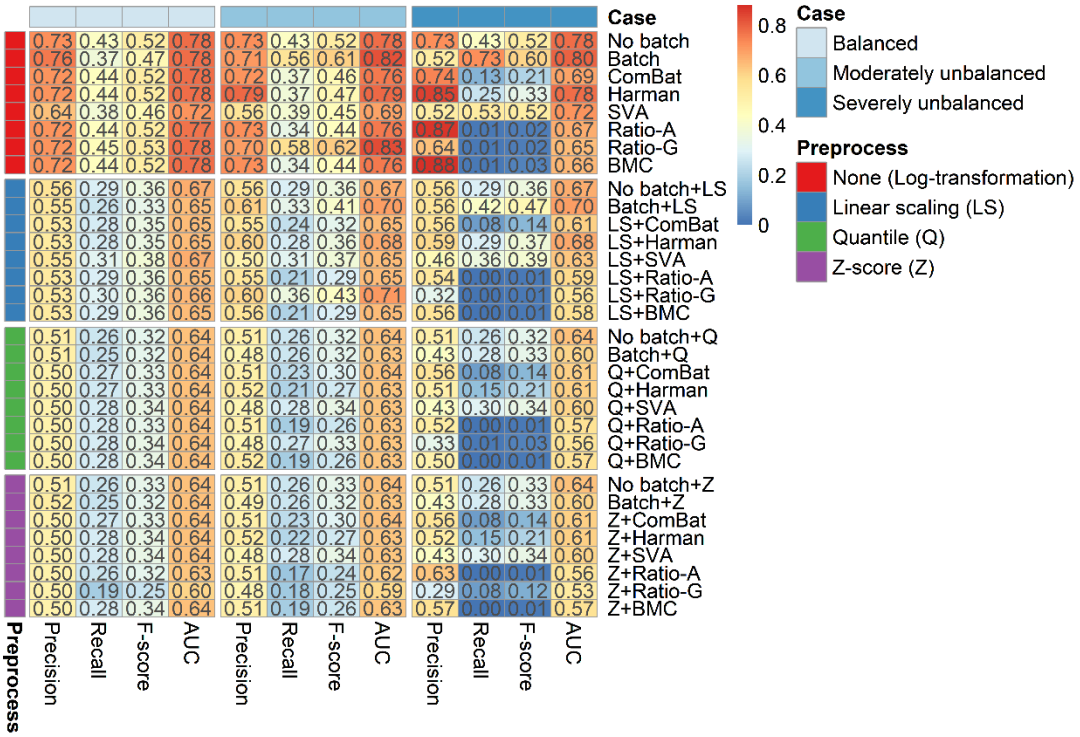


Fig. B27. The impact of prior normalization on BECAs (RNA-seq data 3, Uniformity assumption for batch-class, 100 simulations)**.**

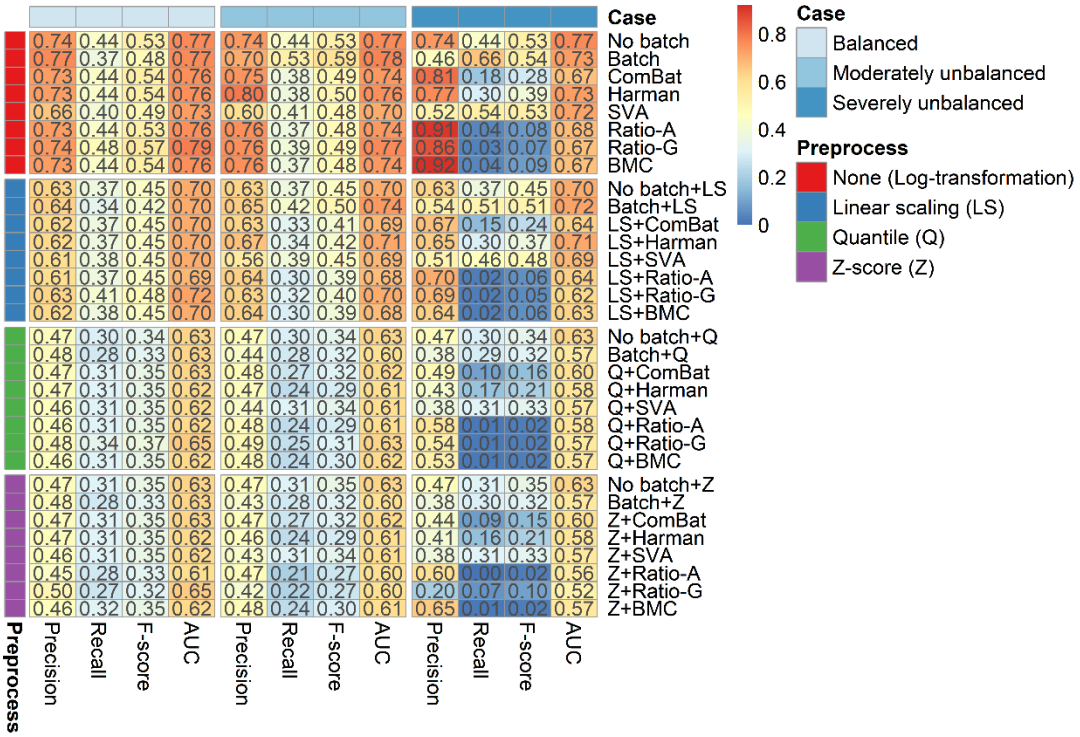


Fig. B28. The impact of prior normalization on BECAs (Proteomics, Uniformity assumption for batch-class, 100 simulations).

Appendix C: More robust batch correction with ComBat in data with batch-class confounding issues

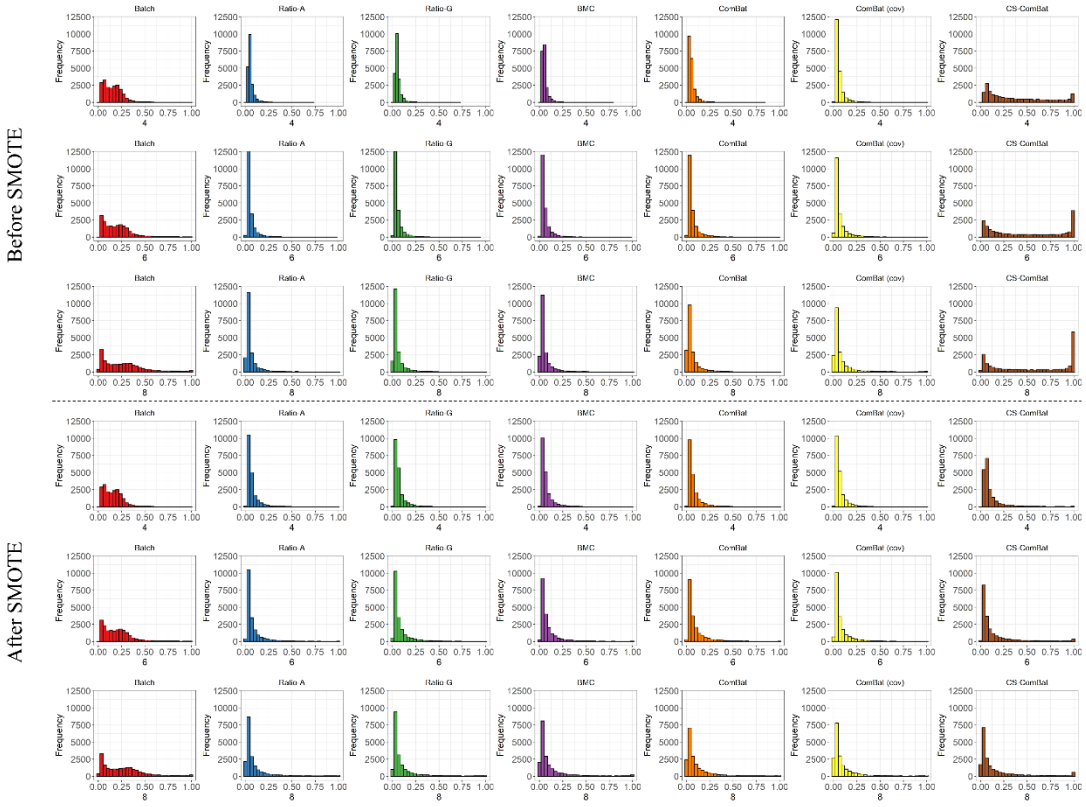


Fig. C1. Performance evaluation based on feature-selection stability of BECAs on real data before SMOTE (**A**) and after SMOTE (**B**). The x-axis shows the sampling size of 4, 6 and 8 of each method. Also, it indicates the normalized value represents feature stability, where 0 means the feature was insignificant across all 1000 simulations, while 1 means the feature was significant across all 1000 simulations. The y-axis shows the frequency value.

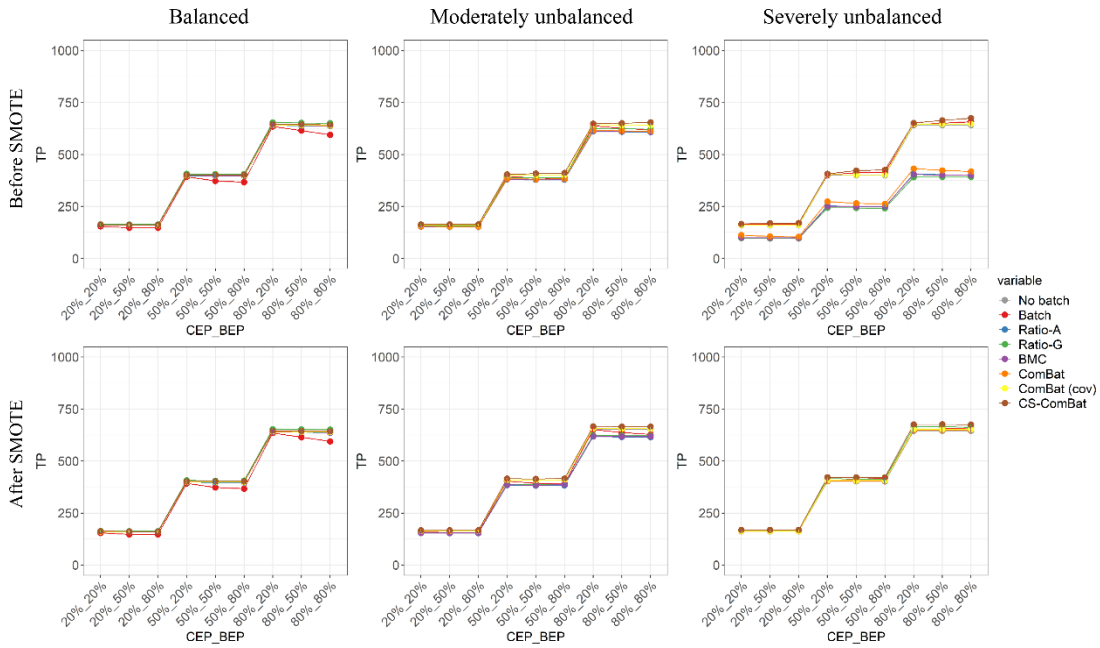


Fig. C2. Performance evaluation based on true positives (TP) of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion (CEP) and batch effects proportion (BEP) ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

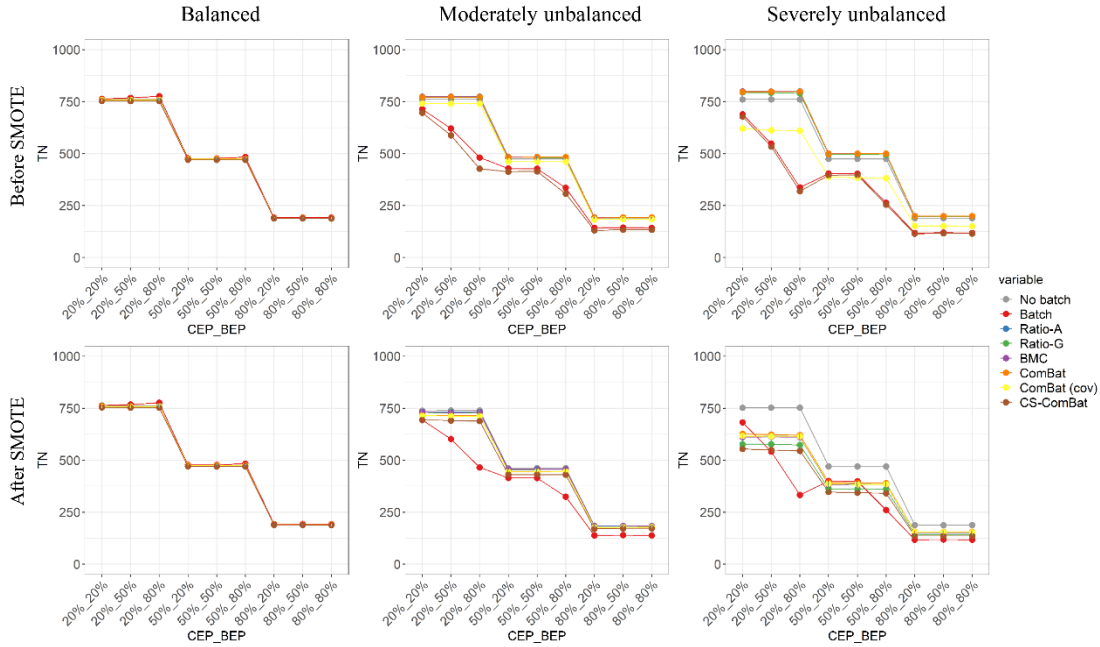


Fig. C3. Performance evaluation based on true negatives (TN) of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion (CEP) and batch effects proportion (BEP) ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

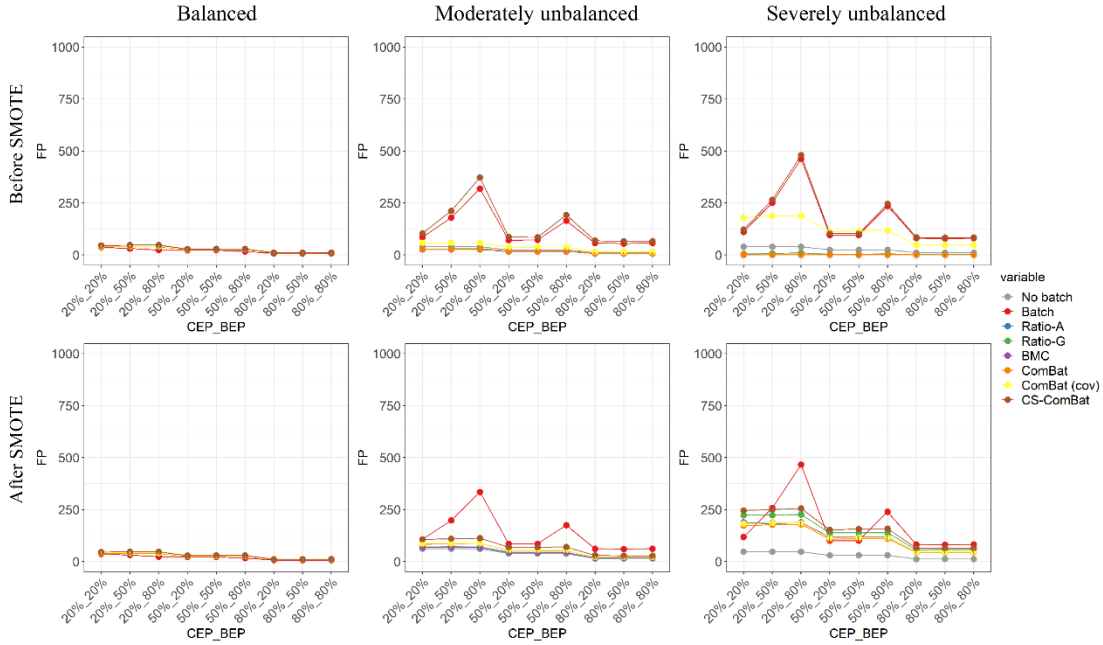


Fig. C4. Performance evaluation based on false positives (FP) of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion (CEP) and batch effects proportion (BEP) ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

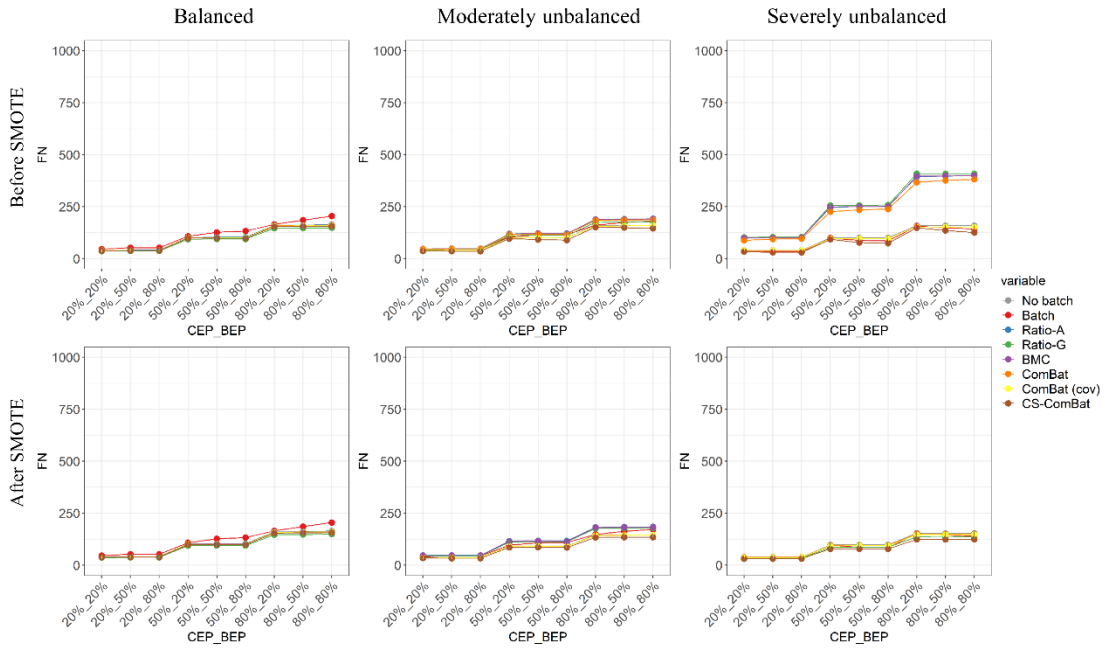


Fig. C5. Performance evaluation based on false negatives (FN) of BECAs on simulated data before SMOTE (**A**) and after SMOTE (**B**). The performance is evaluated under combinations of different degrees of confounding ("balanced," "moderately unbalanced," and "severely unbalanced"), class effects proportion (CEP) and batch effects proportion (BEP) ("20%-low," "50%-medium," and "80%-high"). Error bar represents standard error of 100 simulations.

Appendix D: Class-specific ComBat for correcting batch effects in high-throughput data with a focus on small sample size

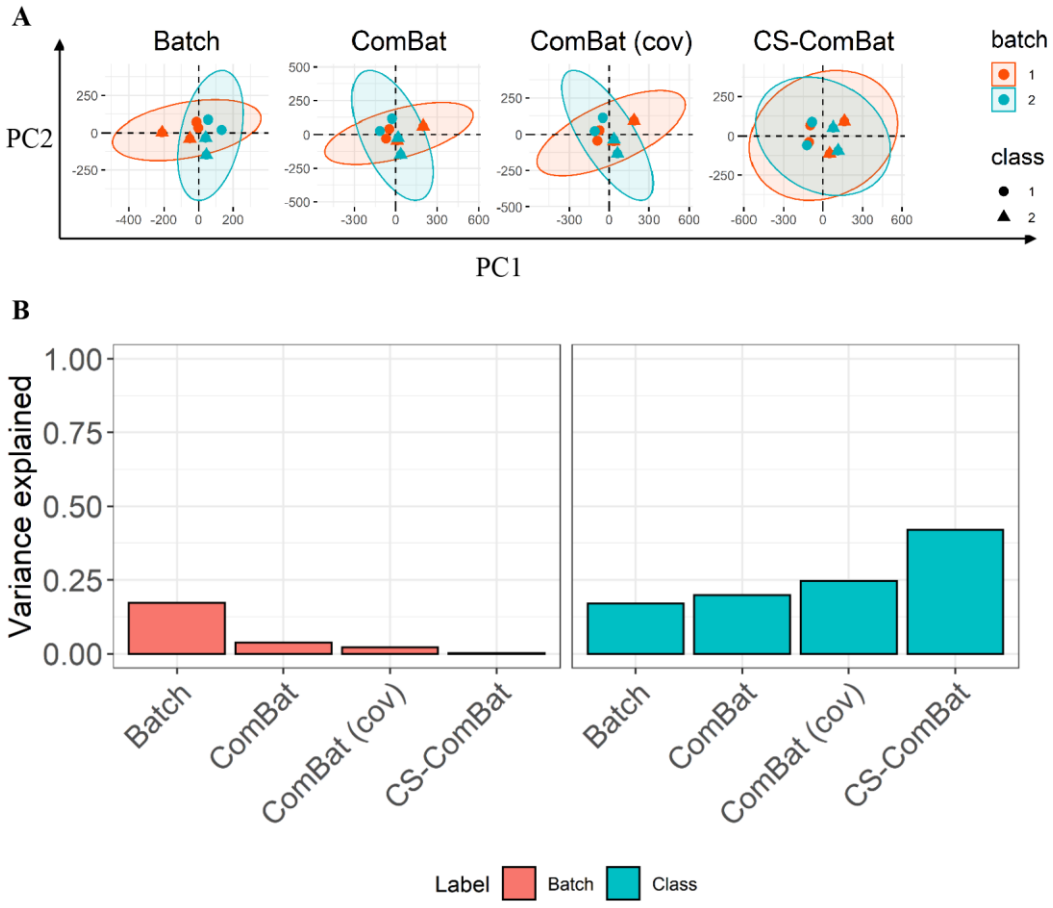


Fig. D1. Batch effects detection in data with real batch effects (NPM. 2H). A. 2D PCA scatterplot plot. B. pRDA results. Batch: data with real batch effects. ComBat, ComBat (cov) and CS-ComBat: batch correction with ComBat, ComBat (cov) and CS-ComBat.

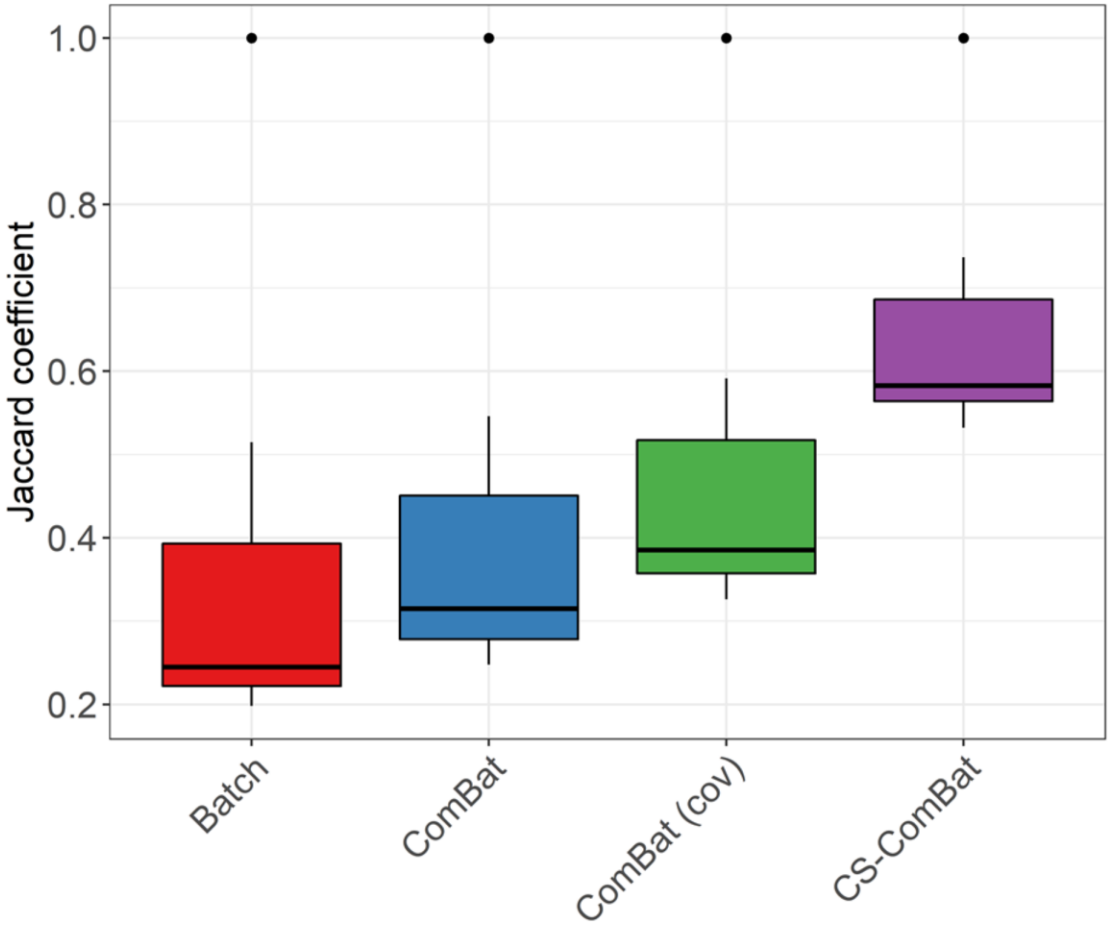


Fig. D2. Inter-sampling similarity evaluation with Jaccard coefficient in data with real batch effects (NPM. 2H). Batch: data with real batch effects. ComBat, ComBat (cov) and CS-ComBat: batch correction with ComBat, ComBat (cov) and CS-ComBat.

Appendix E: Batch effect correction on proteomics data: correcting at peptide level or protein level?

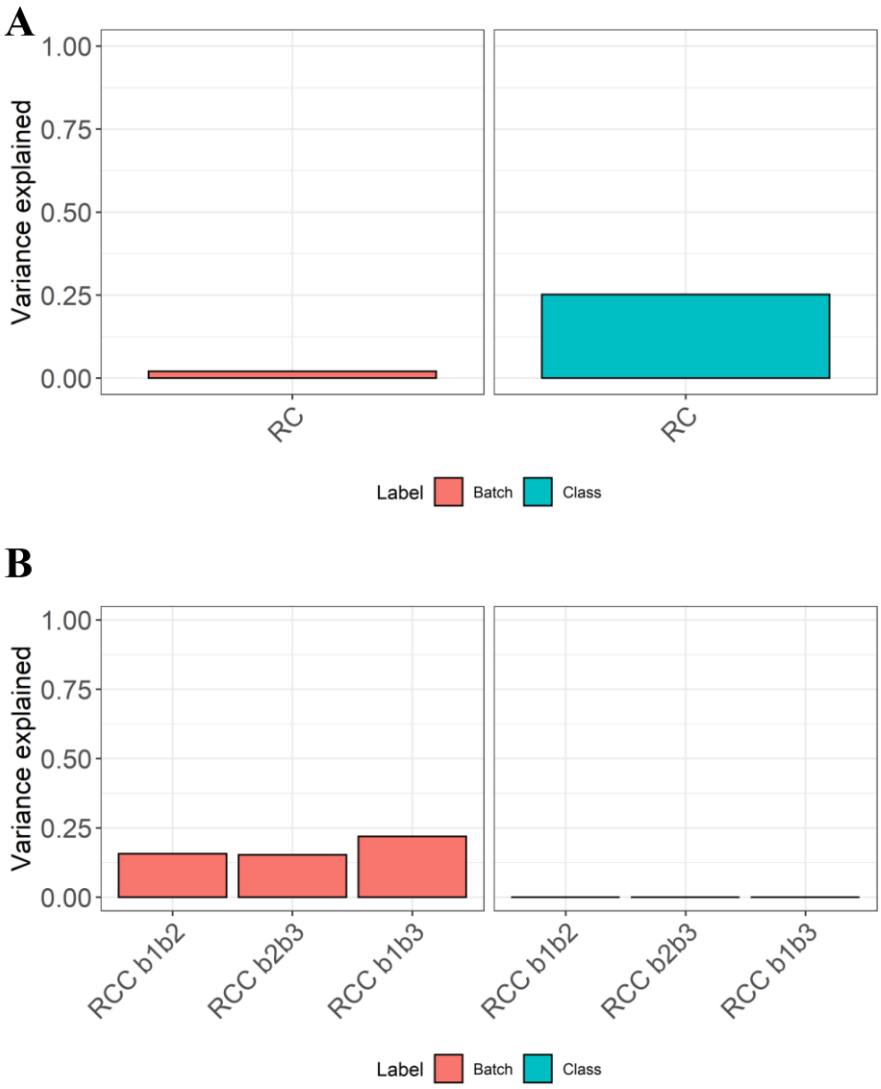


Fig. E1. Batch effects detection with pRDA. A. results of RC data. B. results of RCC data. RCC b1b2: RCC batch 1 and batch 2; RCC b2b3: RCC batch 2 and batch 3; RCC b1b3: RCC batch 1 and batch 3.

Notes on publications and participation in scientific research

List of Publications:

1. First author

[1] **Zhou, L.**; Sue, A. C.-H.; Goh, W. W. B., Examining the practical limits of batch effect-correction algorithms: When should you care about batch effects? *Journal of Genetics and Genomics* **2019,** *46* (9), 433-443.

[2] **Zhou, L**.; Wong, L.; Goh, W. W. B., Understanding missing proteins: a functional perspective. *Drug discovery today* **2018,** *23* (3), 644-651.

2. Second author

[1] Li, H.; **Zhou, L**.; Dai, J., Retinoic acid receptor‐related orphan receptor ROR α regulates differentiation and survival of keratinocytes during hypoxia. *Journal of cellular physiology* **2017,** *233* (1), 641-650.

Acknowledgements

I am deeply grateful to Prof. Wilson Wen Bin Goh for his guidance, for teaching me (1) instill love for scientific research, (2) problem-driven, and (3) critical thinking. Taking me into the field of bioinformatics and let me experience the fun. Giving me valuable advice when I encounter difficulties and frustrations. Without his excellent supervision, this thesis would be impossible to complete. Thanks very much!

I am grateful to Prof. Andrew Chi-Hau Sue for encouraging me when I am lost and giving me support in my research life.

I am grateful to Prof. Limsoon Wong for giving me valuable advice for my paper, make it more solid and persuasive.

I am grateful to Prof. Kim Baldridge, Prof. Adélia Aquino, Prof. Hans Lischka and Prof. Fei Guo for participating in the training of my doctorate and give me valuable experience and insights.

I am grateful to Prof. Jun Dai for training me in cell biology and molecular biology, and providing me valuable suggestion in my life. I am grateful to Prof. Youcai Zhang, Prof. Robert P. Borris and Prof. Haixia Chen for offering me valuable advice.

I am grateful to Yaxing Zhao for learning from each other and growing together.

I am grateful to Wei Xin Chan for helpful discussion.

I am grateful to my beloved Zhongqin Chen for understanding, encouraging and fighting together, and for sharing my happiness and sorrows.

I am grateful to School of Pharmaceutical Science and Technology of Tianjin University for providing a good platform for my study.

I am grateful to my parents for their upbringing, for their supporting me in doing what I love to do.

I am grateful to those people who helped me in my growth. Thanks again!

References

[1] Sachs, M. C. Statistical principles for omics-based clinical trials. *Chin Clin Oncol* **2015,** *4*, 29-39.

[2] Reuter, J. A.; Spacek, D. V.; Snyder, M. P. High-throughput sequencing technologies. *Molecular cell* **2015,** *58*, 586-597.

[3] Soon, W. W.; Hariharan, M.; Snyder, M. P. High-throughput sequencing for biology and medicine. *Molecular systems biology* **2013,** *9*, 640-643.

[4] Van Dijk, E. L.; Auger, H.; Jaszczyszyn, Y.; Thermes, C. Ten years of next-generation sequencing technology. *Trends in genetics : TIG* **2014,** *30*, 418-426.

[5] Goh, W. W. B.; Wang, W.; Wong, L. Why Batch Effects Matter in Omics Data, and How to Avoid Them. *Trends Biotechnol* **2017,** *35*, 498-507.

[6] Kim, S.; Dougherty, E. R.; Chen, Y.; Sivakumar, K.; Meltzer, P.; Trent, J. M.; Bittner, M. Multivariate measurement of gene expression relationships. *Genomics* **2000,** *67*, 201-209.

[7] Duggan, D. J.; Bittner, M.; Chen, Y.; Meltzer, P.; Trent, J. M. Expression profiling using cDNA microarrays. *Nature genetics* **1999,** *21*, 10-14.

[8] Muro, S.; Takemasa, I.; Oba, S.; Matoba, R.; Ueno, N.; Maruyama, C.; Yamashita, R.; Sekimoto, M.; Yamamoto, H.; Nakamori, S.; Monden, M.; Ishii, S.; Kato, K. Identification of expressed genes linked to malignancy of human colorectal carcinoma by parametric clustering of quantitative expression data. *Genome Biol* **2003,** *4*, 1-8.

[9] Kussmann, M.; Raymond, F.; Affolter, M. OMICS-driven biomarker discovery in nutrition and health. *J Biotechnol* **2006,** *124*, 758-787.

[10] Heidecker, B.; Hare, J. M. The use of transcriptomic biomarkers for personalized medicine. *Heart Failure Reviews* **2007,** *12*, 1-11.

[11] Chan, I. S.; Ginsburg, G. S. Personalized medicine: progress and promise. *Annual review of genomics and human genetics* **2011,** *12*, 217-244.

[12] Goh, W. W.; Wong, L. Protein complex-based analysis is resistant to the obfuscating consequences of batch effects --- a case study in clinical proteomics. *BMC Genomics* **2017,** *18*, 142-156.

[13] Benito, M.; Parker, J.; Du, Q.; Wu, J.; Xiang, D.; Perou, C. M.; Marron, J. S. Adjustment of systematic microarray data biases. *Bioinformatics* **2004,** *20*, 105-114.

[14] Johnson, W. E.; Li, C.; Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* **2007,** *8*, 118-127.

[15] Luo, J.; Schumacher, M.; Scherer, A.; Sanoudou, D.; Megherbi, D.; Davison, T.; Shi, T.; Tong, W.; Shi, L.; Hong, H.; Zhao, C.; Elloumi, F.; Shi, W.; Thomas, R.; Lin, S.; Tillinghast, G.; Liu, G.; Zhou, Y.; Herman, D.; Li, Y.; Deng, Y.; Fang, H.; Bushel, P.; Woods, M.; Zhang, J. A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *The pharmacogenomics journal* **2010,** *10*, 278-291.

[16] Lazar, C.; Meganck, S.; Taminau, J.; Steenhoff, D.; Coletta, A.; Molter, C.; Weiss-Solis, D. Y.; Duque, R.; Bersini, H.; Nowe, A. Batch effect removal methods for microarray gene expression data integration: a survey. *Brief Bioinform* **2013,** *14*, 469-490.

[17] Leek, J. T.; Scharpf, R. B.; Bravo, H. C.; Simcha, D.; Langmead, B.; Johnson, W. E.; Geman, D.; Baggerly, K.; Irizarry, R. A. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature reviews. Genetics* **2010,** *11*, 733-739.

[18] Jaksik, R.; Iwanaszko, M.; Rzeszowska-Wolny, J.; Kimmel, M. Microarray experiments and factors which affect their reliability. *Biology direct* **2015,** *10*, 46-59.

[19] Harrison, A.; Binder, H.; Buhot, A.; Burden, C. J.; Carlon, E.; Gibas, C.; Gamble, L. J.; Halperin, A.; Hooyberghs, J.; Kreil, D. P.; Levicky, R.; Noble, P. A.; Ott, A.; Pettitt, B. M.; Tautz, D.; Pozhitkov, A. E. Physico-chemical foundations underpinning microarray and next-generation sequencing experiments. *Nucleic Acids Res* **2013,** *41*, 2779-2796.

[20] Genomes Project, C.; Abecasis, G. R.; Auton, A.; Brooks, L. D.; DePristo, M. A.; Durbin, R. M.; Handsaker, R. E.; Kang, H. M.; Marth, G. T.; McVean, G. A. An integrated map of genetic variation from 1,092 human genomes. *Nature* **2012,** *491*, 56-65.

[21] Lambert, C. G.; Black, L. J. Learning from our GWAS mistakes: from experimental design to scientific method. *Biostatistics* **2012,** *13*, 195-203.

[22] Akey, J. M.; Biswas, S.; Leek, J. T.; Storey, J. D. On the design and analysis of gene expression studies in human populations. *Nat Genet* **2007,** *39*, 807-808.

[23] Cusanovich, D. A.; Pavlovic, B.; Pritchard, J. K.; Gilad, Y. The functional consequences of variation in transcription factor binding. *PLoS Genet* **2014,** *10*, e1004226.

[24] Edgar, R.; Domrachev, M.; Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res* **2002,** *30*, 207-210.

[25] Kolesnikov, N.; Hastings, E.; Keays, M.; Melnichuk, O.; Tang, Y. A.; Williams, E.; Dylag, M.; Kurbatova, N.; Brandizi, M.; Burdett, T.; Megy, K.; Pilicheva, E.; Rustici, G.; Tikhonov, A.; Parkinson, H.; Petryszak,

R.; Sarkans, U.; Brazma, A. ArrayExpress update—simplifying data submissions. *Nucleic Acids Research* **2014,** *43*, 1113-1116.

[26] Cancer Genome Atlas Research, N.; Weinstein, J. N.; Collisson, E. A.; Mills, G. B.; Shaw, K. R.; Ozenberger, B. A.; Ellrott, K.; Shmulevich, I.; Sander, C.; Stuart, J. M. The Cancer Genome Atlas Pan-Cancer analysis project. *Nat Genet* **2013,** *45*, 1113-1120.

[27] Rhodes, D. R.; Yu, J.; Shanker, K.; Deshpande, N.; Varambally, R.; Ghosh, D.; Barrette, T.; Pandey, A.; Chinnaiyan, A. M. ONCOMINE: a cancer microarray database and integrated data-mining platform. *Neoplasia* **2004,** *6*, 1-6.

[28] Consortium, E. P. An integrated encyclopedia of DNA elements in the human genome. *Nature* **2012,** *489*, 57-74.

[29] Yang, W.; Soares, J.; Greninger, P.; Edelman, E. J.; Lightfoot, H.; Forbes, S.; Bindal, N.; Beare, D.; Smith, J. A.; Thompson, I. R.; Ramaswamy, S.; Futreal, P. A.; Haber, D. A.; Stratton, M. R.; Benes, C.; McDermott, U.; Garnett, M. J. Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res* **2013,** *41*, 955-961.

[30] Consortium, G. T.; Laboratory, D. A.; Coordinating Center -Analysis Working, G.; Statistical Methods groups-Analysis Working, G.; Enhancing, G. g.; Fund, N. I. H. C.; Nih/Nci; Nih/Nhgri; Nih/Nimh; Nih/Nida; Biospecimen Collection Source Site, N.; Biospecimen Collection Source Site, R.; Biospecimen Core Resource, V.; Brain Bank Repository-University of Miami Brain Endowment, B.; Leidos Biomedical-Project, M.; Study, E.; Genome Browser Data, I.; Visualization, E. B. I.; Genome Browser Data, I.; Visualization-Ucsc Genomics Institute, U. o. C. S. C.; Lead, a.; Laboratory, D. A.; Coordinating, C.; management, N. I. H. p.; Biospecimen, c.; Pathology; e, Q. T. L. m. w. g.; Battle, A.; Brown, C. D.; Engelhardt, B. E.; Montgomery, S. B. Genetic effects on gene expression across human tissues. *Nature* **2017,** *550*, 204-213.

[31] Stunnenberg, H. G.; International Human Epigenome, C.; Hirst, M. The International Human Epigenome Consortium: A Blueprint for Scientific Collaboration and Discovery. *Cell* **2016,** *167*, 1145-1149.

[32] UniProt Consortium, T. UniProt: the universal protein knowledgebase. *Nucleic Acids Res* **2018,** *46*, 158-169.

[33] Lee, K.-M.; Kim, J.-H.; Kang, D. Design issues in toxicogenomics using DNA microarray experiment. *Toxicology and applied pharmacology* **2005,** *207*, 200-208.

[34] Yang, Y. H.; Speed, T. Design issues for cDNA microarray experiments. *Nature reviews. Genetics* **2002,** *3*, 579-588.

[35] Jaffe, A. E.; Hyde, T.; Kleinman, J.; Weinbergern, D. R.; Chenoweth, J. G.; McKay, R. D.; Leek, J. T.; Colantuoni, C. Practical impacts of

genomic data “cleaning” on biological discovery using surrogate variable analysis. *BMC bioinformatics* **2015,** *16*, 372-381.

[36] Langille, M. G.; Meehan, C. J.; Koenig, J. E.; Dhanani, A. S.; Rose, R. A.; Howlett, S. E.; Beiko, R. G. Microbial shifts in the aging mouse gut. *Microbiome* **2014,** *2*, 1-12.

[37] Wang, Y.; LêCao, K.-A. Managing batch effects in microbiome data. *Briefings in bioinformatics* **2020,** *21*, 1954-1970.

[38] Sims, D.; Sudbery, I.; Ilott, N. E.; Heger, A.; Ponting, C. P. Sequencing depth and coverage: key considerations in genomic analyses. *Nature reviews. Genetics* **2014,** *15*, 121-132.

[39] Huang, J.; Qi, R.; Quackenbush, J.; Dauway, E.; Lazaridis, E.; Yeatman, T. Effects of ischemia on gene expression. *J Surg Res* **2001,** *99*, 222-227.

[40] Katz, S.; Irizarry, R. A.; Lin, X.; Tripputi, M.; Porter, M. W. A summarization approach for Affymetrix GeneChip data using a reference training set from a large, biologically diverse database. *BMC bioinformatics* **2006,** *7*, 464-474.

[41] Thompson, K. L.; Pine, P. S.; Rosenzweig, B. A.; Turpaz, Y.; Retief, J. Characterization of the effect of sample quality on high density oligonucleotide microarray data using progressively degraded rat liver RNA. *BMC biotechnology* **2007,** *7*, 57-68.

[42] Scherer, A. *Batch effects and noise in microarray experiments: sources and solutions*. John Wiley & Sons: 2009; Vol. 868.

[43] Ransohoff, D. F. Bias as a threat to the validity of cancer molecular-marker research. *Nature Reviews Cancer* **2005,** *5*, 142-149.

[44] Čuklina, J.; Pedrioli, P. G.; Aebersold, R. Review of Batch Effects Prevention, Diagnostics, and Correction Approaches. In *Mass Spectrometry Data Analysis in Proteomics*, Springer: 2020; pp 373-387.

[45] Oberg, A. L.; Vitek, O. Statistical design of quantitative mass spectrometry-based proteomic experiments. *J Proteome Res* **2009,** *8*, 2144-2156.

[46] Krzywinski, M.; Altman, N. Points of significance: Analysis of variance and blocking. *Nat Methods* **2014,** *11*, 699-700.

[47] Hu, J.; Coombes, K. R.; Morris, J. S.; Baggerly, K. A. The importance of experimental design in proteomic mass spectrometry experiments: some cautionary tales. *Briefings in Functional Genomics* **2005,** *3*, 322-331.

[48] Gilad, Y.; Mizrahi-Man, O. A reanalysis of mouse ENCODE comparative gene expression data. *F1000Res* **2015,** *4*, 121-152.

[49] Altman, N.; Krzywinski, M. Points of significance: P values and the search for significance. *Nature Methods* **2017,** *14*, 3-4.

[50] Krzywinski, M.; Altman, N. Significance, P values and t-tests: the P value reported by tests is a probabilistic significance, not a biological one. *Nature Methods* **2013,** *10*, 1041-1043.

[51] Blainey, P.; Krzywinski, M.; Altman, N. Points of significance: replication. *Nat Methods* **2014,** *11*, 879-80.

[52] Krzywinski, M.; Altman, N. Points of significance: Power and sample size. *Nature Methods* **2013,** *10*, 1139-1140.

[53] Altman, N.; Krzywinski, M. Points of significance: Sources of variation. *Nat Methods* **2015,** *12*, 5-6.

[54] Skates, S. J.; Gillette, M. A.; LaBaer, J.; Carr, S. A.; Anderson, L.; Liebler, D. C.; Ransohoff, D.; Rifai, N.; Kondratovich, M.; Tezak, Z.; Mansfield, E.; Oberg, A. L.; Wright, I.; Barnes, G.; Gail, M.; Mesri, M.; Kinsinger, C. R.; Rodriguez, H.; Boja, E. S. Statistical design for biospecimen cohort size in proteomics-based biomarker discovery and verification studies. *J Proteome Res* **2013,** *12*, 5383-5394.

[55] Freue, G. V. C.; Meredith, A.; Smith, D.; Bergman, A.; Sasaki, M.; Lam, K. K.; Hollander, Z.; Opushneva, N.; Takhar, M.; Lin, D. Computational biomarker pipeline from discovery to clinical implementation: plasma proteomic biomarkers for cardiac transplantation. *PLoS computational biology* **2013,** *9*, e1002963.

[56] Williams, R. B.; Cotsapas, C. J.; Cowley, M. J.; Chan, E.; Nott, D. J.; Little, P. F. Normalization procedures and detection of linkage signal in genetical-genomics experiments. *Nat Genet* **2006,** *38*, 855-856.

[57] Belorkar, A.; Wong, L. GFS: fuzzy preprocessing for effective gene expression analysis. *BMC Bioinformatics* **2016,** *17*, 540-555.

[58] Goh, W. W. B.; Wong, L. Class-paired Fuzzy SubNETs: A paired variant of the rank-based network analysis family for feature selection based on protein complexes. *Proteomics* **2017,** *17*, e1700093.

[59] Sims, A. H.; Smethurst, G. J.; Hey, Y.; Okoniewski, M. J.; Pepper, S. D.; Howell, A.; Miller, C. J.; Clarke, R. B. The removal of multiplicative, systematic bias allows integration of breast cancer gene expression datasets - improving meta-analysis and prediction of prognosis. *BMC Med Genomics* **2008,** *1*, 42-55.

[60] Leek, J. T.; Storey, J. D. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet* **2007,** *3*, 1724-1735.

[61] Gagnon-Bartsch, J. A.; Speed, T. P. Using control genes to correct for unwanted variation in microarray data. *Biostatistics* **2012,** *13*, 539-552.

[62] Oytam, Y.; Sobhanmanesh, F.; Duesing, K.; Bowden, J. C.; Osmond-McLeod, M.; Ross, J. Risk-conscious correction of batch effects: maximising information extraction from high-throughput genomic datasets. *BMC Bioinformatics* **2016,** *17*, 332-348.

[63] Hornung, R.; Boulesteix, A. L.; Causeur, D. Combining location-and-scale batch effect adjustment with data cleaning by latent factor adjustment. *BMC Bioinformatics* **2016,** *17*, 27-45.

[64] Sims, A. H.; Smethurst, G. J.; Hey, Y.; Okoniewski, M. J.; Pepper, S. D.; Howell, A.; Miller, C. J.; Clarke, R. B. The removal of multiplicative, systematic bias allows integration of breast cancer gene expression datasets–improving meta-analysis and prediction of prognosis. *BMC medical genomics* **2008,** *1*, 42-55.

[65] Stein, C. K.; Qu, P.; Epstein, J.; Buros, A.; Rosenthal, A.; Crowley, J.; Morgan, G.; Barlogie, B. Removing batch effects from purified plasma cell gene expression microarrays with modified ComBat. *BMC Bioinformatics* **2015,** *16*, 63.

[66] Zhang, Y.; Jenkins, D. F.; Manimaran, S.; Johnson, W. E. Alternative empirical Bayes models for adjusting for batch effects in genomic studies. *BMC Bioinformatics* **2018,** *19*, 262-276.

[67] Zhu, T.; Sun, R.; Zhang, F.; Chen, G. B.; Yi, X.; Ruan, G.; Yuan, C.; Zhou, S.; Guo, T. BatchServer: A Web Server for Batch Effect Evaluation, Visualization, and Correction. *J Proteome Res* **2021,** *20*, 1079-1086.

[68] Consortium, E. P. The ENCODE (ENCyclopedia of DNA elements) project. *Science* **2004,** *306*, 636-640.

[69] Consortium, G. T. Human genomics. The Genotype-Tissue Expression (GTEx) pilot analysis: multitissue gene regulation in humans. *Science* **2015,** *348*, 648-660.

[70] Yi, H.; Raman, A. T.; Zhang, H.; Allen, G. I.; Liu, Z. Detecting hidden batch factors through data-adaptive adjustment for biological effects. *Bioinformatics* **2018,** *34*, 1141-1147.

[71] Alter, O.; Brown, P. O.; Botstein, D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci U S A* **2000,** *97*, 10101-10106.

[72] Jolliffe, I. Principal component analysis. *Technometrics* **2003,** *45*, 276.

[73] Giuliani, A. The application of principal component analysis to drug discovery and biomedical data. *Drug Discov Today* **2017,** *22*, 1069-1076.

[74] Goh, W. W. B.; Sng, J. C.-G.; Yee, J. Y.; See, Y. M.; Lee, T.-S.; Wong, L.; Lee, J. Can peripheral blood-derived gene expressions characterize individuals at ultra-high risk for psychosis? *Computational Psychiatry* **2017,** *1*, 168-183.

[75] Parker, H. S.; Leek, J. T.; Favorov, A. V.; Considine, M.; Xia, X.; Chavan, S.; Chung, C. H.; Fertig, E. J. Preserving biological heterogeneity with a permuted surrogate variable analysis for genomics batch correction. *Bioinformatics* **2014,** *30*, 2757-2763.

[76] Leek, J. T. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res* **2014,** *42*, e161.

[77] Chakraborty, S. Use of partial least squares improves the efficacy of removing unwanted variability in differential expression analyses based on RNA-Seq data. *Genomics* **2019,** *111*, 893-898.

[78] Parker, H. S.; Corrada Bravo, H.; Leek, J. T. Removing batch effects for prediction problems with frozen surrogate variable analysis. *PeerJ* **2014,** *2*, e561.

[79] Risso, D.; Ngai, J.; Speed, T. P.; Dudoit, S. Normalization of RNA-seq data using factor analysis of control genes or samples. *Nature biotechnology* **2014,** *32*, 896-905.

[80] Hornung, R.; Boulesteix, A. L.; Causeur, D. Combining location-and-scale batch effect adjustment with data cleaning by latent factor adjustment. *BMC Bioinformatics* **2016,** *17*, 27-45.

[81] Goh, W. W. B.; Wong, L. Dealing with Confounders in Omics Analysis. *Trends Biotechnol* **2018,** *36*, 488-498.

[82] Gandolfo, L. C.; Speed, T. P. RLE plots: Visualizing unwanted variation in high dimensional data. *PLoS One* **2018,** *13*, e0191629.

[83] Papiez, A.; Marczyk, M.; Polanska, J.; Polanski, A. BatchI: Batch effect Identification in high-throughput screening data using a dynamic programming algorithm. *Bioinformatics* **2019,** *35*, 1885-1892.

[84] Chi, E. C.; Lange, K. Splitting Methods for Convex Clustering. *J Comput Graph Stat* **2015,** *24*, 994-1013.

[85] Ding, C. H.; Li, T.; Jordan, M. I. Convex and semi-nonnegative matrix factorizations. *IEEE transactions on pattern analysis and machine intelligence* **2008,** *32*, 45-55.

[86] Jackson, B.; Scargle, J. D.; Barnes, D.; Arabhi, S.; Alt, A.; Gioumousis, P.; Gwin, E.; Sangtrakulcharoen, P.; Tan, L.; Tsai, T. T. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters* **2005,** *12*, 105-108.

[87] Reese, S. E.; Archer, K. J.; Therneau, T. M.; Atkinson, E. J.; Vachon, C. M.; De Andrade, M.; Kocher, J.-P. A.; Eckel-Passow, J. E. A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. *Bioinformatics* **2013,** *29*, 2877-2883.

[88] Nyamundanda, G.; Poudel, P.; Patil, Y.; Sadanandam, A. A Novel Statistical Method to Diagnose, Quantify and Correct Batch Effects in Genomic Studies. *Sci Rep* **2017,** *7*, 1-10.

[89] Kim, K. Y.; Kim, S. H.; Ki, D. H.; Jeong, J.; Jeong, H. J.; Jeung, H. C.; Chung, H. C.; Rha, S. Y. An attempt for combining microarray data sets by adjusting gene expressions. *Cancer Res Treat* **2007,** *39*, 74-81.

[90] Lin, Y.; Ghazanfar, S.; Wang, K. Y. X.; Gagnon-Bartsch, J. A.; Lo, K. K.; Su, X.; Han, Z. G.; Ormerod, J. T.; Speed, T. P.; Yang, P.; Yang, J. Y. H. scMerge leverages factor analysis, stable expression, and pseudoreplication to

merge multiple single-cell RNA-seq datasets. *Proc Natl Acad Sci U S A* **2019,** *116*, 9775-9784.

[91] Florian, S.; Markus, L.; Engin, C.; Sebastian, K.; Jonathan, G.; Marcel, H. S. An ontology-based method for assessing batch effect adjustment approaches in heterogeneous datasets. *Bioinformatics* **2018**, 908-916.

[92] Borcard, D.; Legendre, P.; Drapeau, P. Partialling out the spatial component of ecological variation. *Ecology* **1992,** *73*, 1045-1055.

[93] Nyamundanda, G.; Poudel, P.; Patil, Y.; Sadanandam, A. A novel statistical method to diagnose, quantify and correct batch effects in genomic studies. *Scientific reports* **2017,** *7*, 1-10.

[94] Zhou, L.; Chi-Hau Sue, A.; Bin Goh, W. W. Examining the practical limits of batch effect-correction algorithms: When should you care about batch effects? *J Genet Genomics* **2019,** *46*, 433-443.

[95] Kim, K. Y.; Ki, D. H.; Jeong, H. J.; Jeung, H. C.; Chung, H. C.; Rha, S. Y. Novel and simple transformation algorithm for combining microarray data sets. *BMC Bioinformatics* **2007,** *8*, 218-229.

[96] Chen, J. J.; Delongchamp, R. R.; Tsai, C. A.; Hsueh, H. M.; Sistare, F.; Thompson, K. L.; Desai, V. G.; Fuscoe, J. C. Analysis of variance components in gene expression data. *Bioinformatics* **2004,** *20*, 1436-1446.

[97] Sahai, K. H. Variance Components Analysis: A Selective Literature Survey. *International Statistical Review* **1985,** *53*, 279-300.

[98] Boedigheimer, M. J.; Wolfinger, R. D.; Bass, M. B.; Bushel, P. R.; Chou, J. W.; Cooper, M.; Corton, J. C.; Fostel, J.; Hester, S.; Lee, J. S.; Liu, F.; Liu, J.; Qian, H. R.; Quackenbush, J.; Pettit, S.; Thompson, K. L. Sources of variation in baseline gene expression levels from toxicogenomics study control animals across multiple laboratories. *BMC Genomics* **2008,** *9*, 285-300.

[99] Chen, C.; Grennan, K.; Badner, J.; Zhang, D.; Gershon, E.; Jin, L.; Liu, C. Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PLoS One* **2011,** *6*, e17238.

[100] Li, X. J.; Lee, L. W.; Hayward, C.; Brusniak, M. Y.; Fong, P. Y.; McLean, M.; Mulligan, J.; Spicer, D.; Fang, K. C.; Hunsucker, S. W.; Kearney, P. An integrated quantification method to increase the precision, robustness, and resolution of protein measurement in human plasma samples. *Clin Proteomics* **2015,** *12*, 3-19.

[101] Olivetti, E.; Greiner, S.; Avesani, P. ADHD diagnosis from multiple data sources with batch effects. *Front Syst Neurosci* **2012,** *6*, 70-79.

[102] Sánchez-Illana, Á.; Piñeiro-Ramos, J. D.; Sanjuan-Herráez, J. D.; Vento, M.; Quintás, G.; Kuligowski, J. Evaluation of batch effect elimination using quality control replicates in LC-MS metabolite profiling. *Analytica chimica acta* **2018,** *1019*, 38-48.

[103] Bevilacqua, V.; Pannarale, P.; Abbrescia, M.; Cava, C.; Paradiso, A.; Tommasi, S. In *Comparison of data-merging methods with SVM attribute selection and classification in breast cancer gene expression*, BMC bioinformatics, BioMed Central: 2012; pp 1-15.

[104] Broadhurst, D.; Goodacre, R.; Reinke, S. N.; Kuligowski, J.; Wilson, I. D.; Lewis, M. R.; Dunn, W. B. Guidelines and considerations for the use of system suitability and quality control samples in mass spectrometry assays applied in untargeted clinical metabolomic studies. *Metabolomics* **2018,** *14*, 72-88.

[105] Sanchez-Illana, A.; Pineiro-Ramos, J. D.; Sanjuan-Herraez, J. D.; Vento, M.; Quintas, G.; Kuligowski, J. Evaluation of batch effect elimination using quality control replicates in LC-MS metabolite profiling. *Anal Chim Acta* **2018,** *1019*, 38-48.

[106] Ferreira, T.; Wilson, S. R.; Choi, Y. G.; Risso, D.; Dudoit, S.; Speed, T. P.; Ngai, J. Silencing of odorant receptor genes by G protein betagamma signaling ensures the expression of one odorant receptor per olfactory sensory neuron. *Neuron* **2014,** *81*, 847-859.

[107] Isogai, Y.; Wu, Z.; Love, M. I.; Ahn, M. H.; Bambah-Mukku, D.; Hua, V.; Farrell, K.; Dulac, C. Multisensory Logic of Infant-Directed Aggression by Males. *Cell* **2018,** *175*, 1827-1841.

[108] Hatzi, K.; Geng, H.; Doane, A. S.; Meydan, C.; LaRiviere, R.; Cardenas, M.; Duy, C.; Shen, H.; Vidal, M. N. C.; Baslan, T.; Mohammad, H. P.; Kruger, R. G.; Shaknovich, R.; Haberman, A. M.; Inghirami, G.; Lowe, S. W.; Melnick, A. M. Histone demethylase LSD1 is required for germinal center formation and BCL6-driven lymphomagenesis. *Nat Immunol* **2019,** *20*, 86-96.

[109] Shao, C.; Liu, Y.; Ruan, H.; Li, Y.; Wang, H.; Kohl, F.; Goropashnaya, A. V.; Fedorov, V. B.; Zeng, R.; Barnes, B. M.; Yan, J. Shotgun proteomics analysis of hibernating arctic ground squirrels. *Mol Cell Proteomics* **2010,** *9*, 313-326.

[110] Frazee, A. C.; Jaffe, A. E.; Langmead, B.; Leek, J. T. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics* **2015,** *31*, 2778-2784.

[111] Langley, S. R.; Mayr, M. Comparative analysis of statistical methods used for detecting differential expression in label-free mass spectrometry proteomics. *Journal of proteomics* **2015,** *129*, 83-92.

[112] Leek, J. T.; Johnson, W. E.; Parker, H. S.; Jaffe, A. E.; Storey, J. D. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics* **2012,** *28*, 882-883.

[113] Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters* **2006,** *27*, 861-874.

[114] Sing, T.; Sander, O.; Beerenwinkel, N.; Lengauer, T. ROCR: visualizing classifier performance in R. *Bioinformatics* **2005,** *21*, 3940-3941.

[115] Robin, X.; Turck, N.; Hainard, A.; Tiberti, N.; Lisacek, F.; Sanchez, J. C.; Muller, M. pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics* **2011,** *12*, 77-84.

[116] Nygaard, V.; Rodland, E. A.; Hovig, E. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics* **2016,** *17*, 29-39.

[117] Goh, W. W.; Wong, L. Evaluating feature-selection stability in next-generation proteomics. *Journal of bioinformatics and computational biology* **2016,** *14*, 35-48.

[118] Wang, W.; Sue, A. C.; Goh, W. W. B. Feature selection in clinical proteomics: with great power comes great reproducibility. *Drug Discov Today* **2017,** *22*, 912-918.

[119] Jaffe, A. E.; Hyde, T.; Kleinman, J.; Weinbergern, D. R.; Chenoweth, J. G.; McKay, R. D.; Leek, J. T.; Colantuoni, C. Practical impacts of genomic data "cleaning" on biological discovery using surrogate variable analysis. *BMC Bioinformatics* **2015,** *16*, 372-381.

[120] Zindler, T.; Frieling, H.; Neyazi, A.; Bleich, S.; Friedel, E. Simulating ComBat: how batch correction can lead to the systematic introduction of false positive results in DNA methylation microarray studies. *BMC Bioinformatics* **2020,** *21*, 271-285.

[121] Price, E. M.; Robinson, W. P. Adjusting for Batch Effects in DNA Methylation Microarray Data, a Lesson Learned. *Front Genet* **2018,** *9*, 83-89.

[122] Soneson, C.; Gerster, S.; Delorenzi, M. Batch effect confounding leads to strong bias in performance estimates obtained by cross-validation. *PLoS One* **2014,** *9*, e100335.

[123] Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; Kegelmeyer, W. P. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **2002,** *16*, 321-357.

[124] Hawthorn, L.; Luce, J.; Stein, L.; Rothschild, J. Integration of transcript expression, copy number and LOH analysis of infiltrating ductal carcinoma of the breast. *BMC Cancer* **2010,** *10*, 460-475.

[125] Karnoub, A. E.; Dash, A. B.; Vo, A. P.; Sullivan, A.; Brooks, M. W.; Bell, G. W.; Richardson, A. L.; Polyak, K.; Tubo, R.; Weinberg, R. A. Mesenchymal stem cells within tumour stroma promote breast cancer metastasis. *Nature* **2007,** *449*, 557-563.

[126] Carvalho, B. S.; Irizarry, R. A. A framework for oligonucleotide microarray preprocessing. *Bioinformatics* **2010,** *26*, 2363-2367.

[127] Zhou, G.; Soufan, O.; Ewald, J.; Hancock, R. E. W.; Basu, N.; Xia, J. NetworkAnalyst 3.0: a visual analytics platform for comprehensive gene expression profiling and meta-analysis. *Nucleic Acids Res* **2019,** *47*, 234-241.

[128] Wang, Y.; LeCao, K. A. Managing batch effects in microbiome data. *Brief Bioinform* **2020,** *21*, 1954-1970.

[129] Dixon, P. VEGAN, a package of R functions for community ecology. *Journal of Vegetation Science* **2003,** *14*, 927-930.

[130] Chang, C. Y.; Hsu, M. T.; Esposito, E. X.; Tseng, Y. J. Oversampling to overcome overfitting: exploring the relationship between data set composition, molecular descriptors, and predictive modeling methods. *Journal of chemical information and modeling* **2013,** *53*, 958-971.

[131] Hao, M.; Wang, Y.; Bryant, S. H. An efficient algorithm coupled with synthetic minority over-sampling technique to classify imbalanced PubChem BioAssay data. *Anal Chim Acta* **2014,** *806*, 117-127.

[132] Parkinson, H.; Sarkans, U.; Shojatalab, M.; Abeygunawardena, N.; Contrino, S.; Coulson, R.; Farne, A.; Garcia Lara, G.; Holloway, E.; Kapushesky, M.; Lilja, P.; Mukherjee, G.; Oezcimen, A.; Rayner, T.; Rocca-Serra, P.; Sharma, A.; Sansone, S.; Brazma, A. ArrayExpress—a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research* **2005,** *33*, 553-555.

[133] Govindarajan, M.; Wohlmuth, C.; Waas, M.; Bernardini, M. Q.; Kislinger, T. High-throughput approaches for precision medicine in high-grade serous ovarian cancer. *J Hematol Oncol* **2020,** *13*, 134-153.

[134] Judes, G.; Rifai, K.; Daures, M.; Dubois, L.; Bignon, Y. J.; Penault-Llorca, F.; Bernard-Gallon, D. High-throughput <<Omics>> technologies: New tools for the study of triple-negative breast cancer. *Cancer Lett* **2016,** *382*, 77-85.

[135] Nguyen, T.; Tagett, R.; Diaz, D.; Draghici, S. A novel approach for data integration and disease subtyping. *Genome Res* **2017,** *27*, 2025-2039.

[136] Meijnikman, A. S.; Gerdes, V. E.; Nieuwdorp, M.; Herrema, H. Evaluating Causality of Gut Microbiota in Obesity and Diabetes in Humans. *Endocr Rev* **2018,** *39*, 133-153.

[137] Uniken Venema, W. T.; Voskuil, M. D.; Dijkstra, G.; Weersma, R. K.; Festen, E. A. The genetic background of inflammatory bowel disease: from correlation to causality. *J Pathol* **2017,** *241*, 146-158.

[138] Bastida, J. M.; Lozano, M. L.; Benito, R.; Janusz, K.; Palma-Barqueros, V.; Del Rey, M.; Hernandez-Sanchez, J. M.; Riesco, S.; Bermejo, N.; Gonzalez-Garcia, H.; Rodriguez-Alen, A.; Aguilar, C.; Sevivas, T.; Lopez-Fernandez, M. F.; Marneth, A. E.; van der Reijden, B. A.; Morgan, N. V.; Watson, S. P.; Vicente, V.; Hernandez-Rivas, J. M.; Rivera, J.; Gonzalez-Porras, J. R. Introducing high-throughput sequencing into mainstream genetic diagnosis practice in inherited platelet disorders. *Haematologica* **2018,** *103*, 148-162.

[139] Bansal, V.; Gassenhuber, J.; Phillips, T.; Oliveira, G.; Harbaugh, R.; Villarasa, N.; Topol, E. J.; Seufferlein, T.; Boehm, B. O. Spectrum of mutations in monogenic diabetes genes identified from high-throughput DNA sequencing of 6888 individuals. *BMC Med* **2017,** *15*, 213-226.

[140] Zeng, Z.; Liu, W.; Tsao, T.; Qiu, Y.; Zhao, Y.; Samudio, I.; Sarbassov, D. D.; Kornblau, S. M.; Baggerly, K. A.; Kantarjian, H. M.; Konopleva, M.; Andreeff, M. High-throughput profiling of signaling networks identifies mechanism-based combination therapy to eliminate microenvironmental resistance in acute myeloid leukemia. *Haematologica* **2017,** *102*, 1537-1548.

[141] Fu, Y.; Sun, Y.; Li, Y.; Li, J.; Rao, X.; Chen, C.; Xu, A. Differential genome-wide profiling of tandem 3' UTRs among human breast cancer and normal cells by high-throughput sequencing. *Genome Res* **2011,** *21*, 741-747.

[142] Shao, L.; Liu, Y.; Mei, J.; Li, D.; Chen, L.; Pan, Q.; Zhang, S.; Dai, X.; Liang, J.; Sun, S.; Wang, J. High-throughput sequencing reveals the diversity of TCR beta chain CDR3 repertoire in patients with severe acne. *Mol Immunol* **2020,** *120*, 23-31.

[143] Ye, B.; Smerin, D.; Gao, Q.; Kang, C.; Xiong, X. High-throughput sequencing of the immune repertoire in oncology: Applications for clinical diagnosis, monitoring, and immunotherapies. *Cancer Lett* **2018,** *416*, 42-56.

[144] Schwarz, U. I.; Gulilat, M.; Kim, R. B. The Role of Next-Generation Sequencing in Pharmacogenetics and Pharmacogenomics. *Cold Spring Harbor perspectives in medicine* **2019,** *9*, 1-16.

[145] Sankaran, V. G.; Gallagher, P. G. Applications of high-throughput DNA sequencing to benign hematology. *Blood* **2013,** *122*, 3575-3582.

[146] Auer, P. L.; Doerge, R. W. Statistical design and analysis of RNA sequencing data. *Genetics* **2010,** *185*, 405-416.

[147] Osmond-McLeod, M. J.; Oytam, Y.; Kirby, J. K.; Gomez-Fernandez, L.; Baxter, B.; McCall, M. J. Dermal absorption and short-term biological impact in hairless mice from sunscreens containing zinc oxide nano- or larger particles. *Nanotoxicology* **2014,** *8* 72-84.

[148] Zhang, Z.; Wu, S.; Stenoien, D. L.; Paša-Tolić, L. High-throughput proteomics. *Annual review of analytical chemistry (Palo Alto, Calif.)* **2014,** *7*, 427-454.

[149] Wilhelm, M.; Schlegl, J.; Hahne, H.; Gholami, A. M.; Lieberenz, M.; Savitski, M. M.; Ziegler, E.; Butzmann, L.; Gessulat, S.; Marx, H.; Mathieson, T.; Lemeer, S.; Schnatbaum, K.; Reimer, U.; Wenschuh, H.; Mollenhauer, M.; Slotta-Huspenina, J.; Boese, J. H.; Bantscheff, M.; Gerstmair, A.; Faerber, F.; Kuster, B. Mass-spectrometry-based draft of the human proteome. *Nature* **2014,** *509*, 582-587.

[150] Kim, M. S.; Pinto, S. M.; Getnet, D.; Nirujogi, R. S.; Manda, S. S.; Chaerkady, R.; Madugundu, A. K.; Kelkar, D. S.; Isserlin, R.; Jain, S.;

Thomas, J. K.; Muthusamy, B.; Leal-Rojas, P.; Kumar, P.; Sahasrabuddhe, N. A.; Balakrishnan, L.; Advani, J.; George, B.; Renuse, S.; Selvan, L. D.; Patil, A. H.; Nanjappa, V.; Radhakrishnan, A.; Prasad, S.; Subbannayya, T.; Raju, R.; Kumar, M.; Sreenivasamurthy, S. K.; Marimuthu, A.; Sathe, G. J.; Chavan, S.; Datta, K. K.; Subbannayya, Y.; Sahu, A.; Yelamanchi, S. D.; Jayaram, S.; Rajagopalan, P.; Sharma, J.; Murthy, K. R.; Syed, N.; Goel, R.; Khan, A. A.; Ahmad, S.; Dey, G.; Mudgal, K.; Chatterjee, A.; Huang, T. C.; Zhong, J.; Wu, X.; Shaw, P. G.; Freed, D.; Zahari, M. S.; Mukherjee, K. K.; Shankar, S.; Mahadevan, A.; Lam, H.; Mitchell, C. J.; Shankar, S. K.; Satishchandra, P.; Schroeder, J. T.; Sirdeshmukh, R.; Maitra, A.; Leach, S. D.; Drake, C. G.; Halushka, M. K.; Prasad, T. S.; Hruban, R. H.; Kerr, C. L.; Bader, G. D.; Iacobuzio-Donahue, C. A.; Gowda, H.; Pandey, A. A draft map of the human proteome. *Nature* **2014,** *509*, 575-581.

[151] Filén, J. J.; Filén, S.; Moulder, R.; Tuomela, S.; Ahlfors, H.; West, A.; Kouvonen, P.; Kantola, S.; Björkman, M.; Katajamaa, M.; Rasool, O.; Nyman, T. A.; Lahesmaa, R. Quantitative proteomics reveals GIMAP family proteins 1 and 4 to be differentially regulated during human T helper cell differentiation. *Mol Cell Proteomics* **2009,** *8*, 32-44.

[152] Zhou, L.; Wong, L.; Goh, W. W. B. Understanding missing proteins: a functional perspective. *Drug Discov Today* **2018,** *23*, 644-651.

[153] Lundberg, E.; Borner, G. H. H. Spatial proteomics: a powerful discovery tool for cell biology. *Nature reviews. Molecular cell biology* **2019,** *20*, 285-302.

[154] Cookson, M. R. Proteomics: techniques and applications in neuroscience. *Journal of neurochemistry* **2019,** *151*, 394-396.

[155] Manes, N. P.; Nita-Lazar, A. Application of targeted mass spectrometry in bottom-up proteomics for systems biology research. *J Proteomics* **2018,** *189*, 75-90.

[156] Li, X.; Wang, W.; Chen, J. Recent progress in mass spectrometry proteomics for biomedical research. *Science China. Life sciences* **2017,** *60*, 1093-1113.

[157] Guo, T.; Kouvonen, P.; Koh, C. C.; Gillet, L. C.; Wolski, W. E.; Röst, H. L.; Rosenberger, G.; Collins, B. C.; Blum, L. C.; Gillessen, S.; Joerger, M.; Jochum, W.; Aebersold, R. Rapid mass spectrometric conversion of tissue biopsy samples into permanent quantitative digital proteome maps. *Nat Med* **2015,** *21*, 407-413.

[158] Gillet, L. C.; Navarro, P.; Tate, S.; Röst, H.; Selevsek, N.; Reiter, L.; Bonner, R.; Aebersold, R. Targeted data extraction of the MS/MS spectra generated by data-independent acquisition: a new concept for consistent and accurate proteome analysis. *Mol Cell Proteomics* **2012,** *11*, 1-17.

[159] Zhu, Y.; Weiss, T.; Zhang, Q.; Sun, R.; Wang, B.; Yi, X.; Wu, Z.; Gao, H.; Cai, X.; Ruan, G.; Zhu, T.; Xu, C.; Lou, S.; Yu, X.; Gillet,

L.; Blattmann, P.; Saba, K.; Fankhauser, C. D.; Schmid, M. B.; Rutishauser, D.; Ljubicic, J.; Christiansen, A.; Fritz, C.; Rupp, N. J.; Poyet, C.; Rushing, E.; Weller, M.; Roth, P.; Haralambieva, E.; Hofer, S.; Chen, C.; Jochum, W.; Gao, X.; Teng, X.; Chen, L.; Zhong, Q.; Wild, P. J.; Aebersold, R.; Guo, T. High-throughput proteomic analysis of FFPE tissue samples facilitates tumor stratification. *Mol Oncol* **2019,** *13*, 2305-2328.

[160] Bouchal, P.; Schubert, O. T.; Faktor, J.; Capkova, L.; Imrichova, H.; Zoufalova, K.; Paralova, V.; Hrstka, R.; Liu, Y.; Ebhardt, H. A.; Budinska, E.; Nenutil, R.; Aebersold, R. Breast Cancer Classification Based on Proteotypes Obtained by SWATH Mass Spectrometry. *Cell reports* **2019,** *28*, 832-843.

[161] Krasny, L.; Bland, P.; Kogata, N.; Wai, P.; Howard, B. A.; Natrajan, R. C.; Huang, P. H. SWATH mass spectrometry as a tool for quantitative profiling of the matrisome. *J Proteomics* **2018,** *189*, 11-22.

[162] Collins, B. C.; Hunter, C. L.; Liu, Y.; Schilling, B.; Rosenberger, G.; Bader, S. L.; Chan, D. W.; Gibson, B. W.; Gingras, A. C.; Held, J. M.; Hirayama-Kurogi, M.; Hou, G.; Krisp, C.; Larsen, B.; Lin, L.; Liu, S.; Molloy, M. P.; Moritz, R. L.; Ohtsuki, S.; Schlapbach, R.; Selevsek, N.; Thomas, S. N.; Tzeng, S. C.; Zhang, H.; Aebersold, R. Multi-laboratory assessment of reproducibility, qualitative and quantitative performance of SWATH-mass spectrometry. *Nat Commun* **2017,** *8*, 291-302.

[163] Narasimhan, M.; Kannan, S.; Chawade, A.; Bhattacharjee, A.; Govekar, R. Clinical biomarker discovery by SWATH-MS based label-free quantitative proteomics: impact of criteria for identification of differentiators and data normalization method. *J Transl Med* **2019,** *17*, 184-198.

[164] Anjo, S. I.; Santa, C.; Manadas, B. SWATH-MS as a tool for biomarker discovery: From basic research to clinical applications. *Proteomics* **2017,** *17*.

[165] López-Sánchez, L. M.; Jiménez-Izquierdo, R.; Peñarando, J.; Mena, R.; Guil-Luna, S.; Toledano, M.; Conde, F.; Villar, C.; Díaz, C.; Ortea, I.; De la Haba-Rodríguez, J. R.; Aranda, E.; Rodríguez-Ariza, A. SWATH-based proteomics reveals processes associated with immune evasion and metastasis in poor prognosis colorectal tumours. *Journal of cellular and molecular medicine* **2019,** *23*, 8219-8232.

[166] Hedl, T. J.; San Gil, R.; Cheng, F.; Rayner, S. L.; Davidson, J. M.; De Luca, A.; Villalva, M. D.; Ecroyd, H.; Walker, A. K.; Lee, A. Proteomics Approaches for Biomarker and Drug Target Discovery in ALS and FTD. *Frontiers in neuroscience* **2019,** *13*, 1-25.

[167] Raimondo, S.; Cristaldi, M.; Fontana, S.; Saieva, L.; Monteleone, F.; Calabrese, G.; Giavaresi, G.; Parenti, R.; Alessandro, R. The phospholipase

DDHD1 as a new target in colorectal cancer therapy. *Journal of experimental & clinical cancer research : CR* **2018,** *37*, 82-93.

[168] Saleh, S.; Staes, A.; Deborggraeve, S.; Gevaert, K. Targeted Proteomics for Studying Pathogenic Bacteria. *Proteomics* **2019,** *19*, e1800435.

[169] Marchat, L. A.; Hernandez-de la Cruz, O. N.; Ramirez-Moreno, E.; Silva-Cazares, M. B.; Lopez-Camarillo, C. Proteomics approaches to understand cell biology and virulence of Entamoeba histolytica protozoan parasite. *J Proteomics* **2020,** *226*, 103897-10396.

[170] Kang, S.; Kong, F.; Liang, X.; Li, M.; Yang, N.; Cao, X.; Yang, M.; Tao, D.; Yue, X.; Zheng, Y. Label-Free Quantitative Proteomics Reveals the Multitargeted Antibacterial Mechanisms of Lactobionic Acid against Methicillin-Resistant Staphylococcus aureus (MRSA) using SWATH-MS Technology. *Journal of agricultural and food chemistry* **2019,** *67*, 12322-12332.

[171] Borrebaeck, C. A. Precision diagnostics: moving towards protein biomarker signatures of clinical utility in cancer. *Nat Rev Cancer* **2017,** *17*, 199-204.

[172] Kowalczyk, T.; Ciborowski, M.; Kisluk, J.; Kretowski, A.; Barbas, C. Mass spectrometry based proteomics and metabolomics in personalized oncology. *Biochim Biophys Acta Mol Basis Dis* **2020,** *1866*, 1-18.

[173] Macklin, A.; Khan, S.; Kislinger, T. Recent advances in mass spectrometry based clinical proteomics: applications to cancer research. *Clin Proteomics* **2020,** *17*, 17-41.

[174] Gregori, J.; Villarreal, L.; Méndez, O.; Sánchez, A.; Baselga, J.; Villanueva, J. Batch effects correction improves the sensitivity of significance tests in spectral counting-based comparative discovery proteomics. *J Proteomics* **2012,** *75*, 3938-3951.

[175] Wang, J.; Wang, P.; Hedeker, D.; Chen, L. S. Using multivariate mixed-effects selection models for analyzing batch-processed proteomics data with non-ignorable missingness. *Biostatistics* **2019,** *20*, 648-665.

[176] Röst, H. L.; Rosenberger, G.; Navarro, P.; Gillet, L.; Miladinović, S. M.; Schubert, O. T.; Wolski, W.; Collins, B. C.; Malmström, J.; Malmström, L.; Aebersold, R. OpenSWATH enables automated, targeted analysis of data-independent acquisition MS data. *Nature biotechnology* **2014,** *32*, 219-223.