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

Zhou, Longjian; Sue, Andrew Chi-Hau; Goh, Wilson Wen Bin

2019

Zhou, L., Sue, A. C. & Goh, W. W. B. (2019). Examining the practical limits of batch effect-correction algorithms : when should you care about batch effects?. Journal of Genetics and Genomics, 46(9), 433-443. <https://dx.doi.org/10.1016/j.jgg.2019.08.002>

<https://hdl.handle.net/10356/150368>

<https://doi.org/10.1016/j.jgg.2019.08.002>

© 2019 Institute of Genetics and Developmental Biology, Chinese Academy of Sciences, and Genetics Society of China. All rights reserved. This paper was published by Elsevier Limited and Science Press in Journal of Genetics and Genomics and is made available with permission of Institute of Genetics and Developmental Biology, Chinese Academy of Sciences, and Genetics Society of China.

Downloaded on 26 Apr 2023 17:54:53 SGT

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

Longjian Zhou¹, Andrew Chi-Hau Sue¹, Wilson Wen Bin Goh^{2*}

1. School of Pharmaceutical Science and Technology, Tianjin University,
Tianjin, 30072, China
2. School of Biological Sciences, Nanyang Technological University, 60
Nanyang Drive, 637551, Singapore

*Corresponding Author: Wilson Wen Bin Goh, wilsongoh@ntu.edu.sg;
goh.informatics@gmail.com

Address for correspondence/proofs:

Wilson Wen Bin Goh, PhD

School of Biological Sciences, Nanyang Technological University, 60 Nanyang
Drive, 637551, Singapore.

Email: wilsongoh@ntu.edu.sg; goh.informatics@gmail.com

Tel: +65-65162902

1 **Abstract**

2 Batch effects are technical sources of variation and can confound analysis. While many
3 performance ranking exercises have been conducted to establish the best Batch effect-
4 correction algorithm (BECA), we hold the viewpoint that the notion of best is context-
5 dependent. Moreover, alternative questions beyond the simplistic notion of “best” are
6 also interesting: are BECAs robust against various degrees of confounding and if so,
7 what is the limit? Using two different methods for simulating class (phenotype) and
8 batch effects and taking various representative datasets across both genomics (RNA-
9 Seq) and proteomics platforms, we demonstrate that up till situations where sample
10 classes and batch factors are moderately confounded, most BECAs are remarkably
11 robust and only weakly affected by upstream normalization procedures. This
12 observation is consistently supported across the multitude of test datasets. BECAs do
13 have limits: When sample classes and batch factors are strongly confounded, BECA
14 performance declines, with variable performance in precision, recall and also batch
15 correction. We also report that while conventional normalization methods have
16 minimal impact on batch effect correction, they do not affect downstream statistical
17 feature selection, and in strongly confounded scenarios, may even outperform BECAs.
18 In other words, removing batch effects is no guarantee of optimal functional analysis.
19 Overall, this study suggests that simplistic performance ranking exercises are quite
20 trivial, all BECAs are a compromise in some context or another.

21 **Keywords**

22 Batch effects; Bioinformatics; Feature selection; Normalization; Statistics

23 **Abbreviations**

24 AUC, area under the curve; BECAs, batch effect-correction algorithms; BEs, Batch
25 effects; BMC, batch mean-centering; gPCA, guided-PCA; PCA, principal components
26 analysis; PCs, principal components; Ratio-A, ratio-based approach (arithmetic mean
27 as reference); Ratio-G, ratio-based approach (geometric mean as reference); ROC,
28 receiver operating characteristic; SVA, surrogate variable analysis.

1. Introduction

Batch effects (BEs) are technical sources of variation arising from different experiment times, handlers, reagents, and instruments (Leek et al., 2010; Goh et al., 2017). BEs are especially relevant to assays with sample-size limits, thereby requiring multiple runs (each run constituting a “technical batch”). BEs are not a homogeneous group of technical variation: Specific assays and platforms can create idiosyncratic batch effects such that within-batch and between-batch effects observed in metabolomics and other related platforms (Broadhurst et al., 2018; Sanchez-Illana et al., 2018). The simulations deployed in this study do not cover these types of BEs as it is very specific, instead we simulate BEs using naïve but more generalizable assumptions as discussed later in the introduction.

BEs may confound functional analysis. In comparative setups where one phenotype class is compared against another (e.g. normal against disease), BEs may induce false effects or mask real effects (Goh and Wong, 2018) and hamper research outcome. Thus, it is important BEs are dealt with satisfactorily.

There is a wide variety of BECAs, necessitating various performance ranking evaluations (Luo et al., 2010; Chen et al., 2011; Lazar et al., 2012). Although ComBat (Johnson et al., 2007), one of the earliest BECAs, often dominates, other methods such as Harman (Oytam et al., 2016), SVA (Leek and Storey, 2007), and the ratio-based algorithms (Luo et al., 2010), Ratio-A, Ratio-G and Batch-mean centering (Sims et al., 2008), 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, amongst 8 samples such that the order of class labels S is $(A_1, A_2, A_3, A_4, B_5, B_6, B_7, B_8)$. 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 $(U_1, U_2, V_3, V_4, U_5, U_6, V_7, V_8)$. Here, although BEs 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, BEs and other sources of technical bias may be “averaged out”. Since BEs are negated, there is no confounding between class and batch factors.

Unfortunately, experimental designs are oftentimes imperfect. Real world factors, e.g. sample availability and failed experiments may also be contributors. These lead towards batch-class factor imbalance, such that BEs cannot be negated via “averaging out”. For example, given $(A_1, A_2, A_3, A_4, B_5, B_6, B_7, B_8)$, we may have $(U_1, U_2, U_3, V_4, U_5, V_6, V_7, V_8)$ 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 $(A_1, A_2, A_3, A_4, B_5, B_6, B_7, B_8)$, we may have $(U_1, U_2, U_3, U_4, V_5, V_6, V_7, V_8)$. 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 BEs 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 varies between different samples e.g. gene or protein expression). Leek et al. suggested that BEs are non-homogeneous, and non-uniformly distributed amongst different variables (Leek et al., 2010). 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 (Goh and Wong, 2017).

This study 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 exist 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 is used for evaluation.

2. Results

2.1 BECAs are surprisingly resilient against batch-class imbalances

2D-scatterplots are a standard for detecting class and batch effects (Fig. 1). We will discuss findings based on one transcriptomics dataset based on the non-uniformity assumption in BE simulation (Genomics 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.

In Genomics 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 effect is present, will not overlap.

PCA shows that the original data clusters 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. 1A) to severely unbalanced batch-class (Fig. 1C).

As expected, when batch-class is balanced (Fig. 1A), BECAs perform well. BECA-adjusted data (Combat, Harman, SVA, Ratio-A, Ratio-G, BMC), shows that samples no longer cluster by batches. However, conventional normalization approaches such as linear scaling, Z-score, and quantile do not remove BEs.

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 BEs. Fig. 1D shows the corresponding gPCA deltas (and its associative *P*-value). From the moderate unbalanced scenario onwards, SVA's gPCA delta moves up dramatically (indicating BEs are not dealt with) suggesting SVA is very sensitive to batch-class imbalance. Note that in Fig. 1B, 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 BEs.

When batch-class design is severely unbalanced, distinguishing BEs from class effects is very challenging. In Fig. 1C, 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, a closer inspection reveal 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: Genomics data 2 and 3, and proteomics data (Fig. S1 to S3), and also when we used a different approach for simulating BEs under the uniformity assumption in the same genomics and proteomics data (Fig. S4 to S7). 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/ do not under any circumstances; rely on conventional normalization to take care of ambient BEs.

2.2 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. Quick get arounds exist in the form of PCA-matrices (Goh et al., 2017) and side-by-side boxplots (Goh and Wong, 2017). But these approaches require expert interpretation. A summary statistic for measuring overall batch effect per simulation is desirable and time-efficient. But does such a thing exist?

Here, we considered the use of the gPCA delta, and its associative *P*-value. In **Fig. 1** (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 BEs in Ratio-A, Ratio-G and BMC (**Fig. 1D**) while also overestimating batch effects in SVA (**Fig. 1D**). But it does overall, capture BEs well (when present). The gPCA *P*-value, on the other hand, is usually significant only when batch effect 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 a batch effect is 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 BEs. Similar results are also observed in Genomics data 2 and 3 and proteomics data (Fig. S1 to S3), and in the uniformity assumption (Fig. S4 to S7).

The representations in Fig. 1 (and Fig. S1 to S7) only convey the results for one simulation. To test for reproducibility and if the gPCA deltas are stable, we apply gPCA across 100 simulated genomics datasets (this is modeled as 100 repetitions of the experiment in Fig. 1). As shown in Fig. 2 (and also in Fig. S8 to S14), the gPCA deltas are largely stable.

As with Fig. 1, 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 BEs. 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 denotes the inter-quartile range) (Fig. 1). This suggests Harman is not likely to perform similarly across different datasets. The genomics data 2 and 3 and proteomics data simulations suggest likewise (Fig. S8 to S10). And this is unlikely due to the batch-class simulation approach. Batch-class simulations using the uniformity assumption also show the same results (Fig. S11 and S14).

Use of the gPCA deltas facilitates rapid detection, quantitation and summarization of BEs across many simulations. The interesting observation is the results are largely stable, irrespective of platform (proteomics or genomics) 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 Seemingly good batch-effect correction is not a reliable indicator of good downstream statistical selection

Fig. 1 (and Fig. S1 to S7) revealed that visual representation and summary statistics for BEs are not always congruent. Resolving BEs is a difficult problem, since BEs are complex, and may not be efficiently estimated and removed, and may produce false positives and false negatives (Goh et al., 2017; Goh and Wong, 2017).

Since the set of true differential genes are 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 badly 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, this is a known problem with SVA (Jaffe et al., 2015).

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 over, 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 or 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 will meet our requirement as a favored BECA (instead of just using a single performance metric).

When batch-class is balanced (Fig. 3A/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 BEs are not removed entirely, it may not adversely affect differential gene identification.

However, performance differentiation quickly emerges from moderate (Fig. 3B/E) to severely unbalanced (Fig. 3C/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 with removing batch effects when batch-class design is unbalanced, does not seem to have any issues with statistical feature selection, be it 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 Genomics data 1 are also supported by our other test datasets (Fig. S15 to S17). The findings are also independent of the batch-class uniformity assumption (Fig. S18 to S21), 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 BEs 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 (Nygaard et al., 2016) observation 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 effect 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 seem 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 to not 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.

2.4 Most BECAs are not particularly sensitive to upstream normalization procedures

Using Fig. 4, with cross-reference against Fig. S22 (Genomics 2), Fig. S23 (Genomics 3), Fig. S24 (Proteomics) and simulations that do not use the batch-class non-uniformity assumptions (Fig. S25 to S28), 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 normalization does not appear to play much impact on performance, it may not be a strong contributing factor for ranking differences across BECA ranking studies. It is possible that the performances differences observed between BECAs may be idiosyncratic to the particular datasets under study, or that ranking is based on very small differences (that we also observe here).

3. Discussion

It is not entirely surprising conventional normalizations perform poorly. After all, they do not explicitly correct for BEs. 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, just because samples look well-standardized does not mean that BEs are corrected for. What QN does 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 BEs.

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 $g > 0$ in class A and true level $g' > 0$ in class B, and $g > 2 * g'$.

Suppose BEs 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 & 25% class B are in the batch, 25% A and 75% B not in the batch), though the classes are balanced (100 samples each).

331 Then the quantile normalized value of gene G in class A is $g * (75\% * 100) * 2 + g$
332 $* (25\% * 100) = g * (0.75 * 2 + 0.25) = 1.75 * g$. And the quantile normalized value of
333 gene G in class B is $g' * 0.25 * 2 + g' * 0.75 = 1.25 * g'$.

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

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

340 Conventional normalization approaches such as Z-Score, linear scaling, and
341 quantile are non-effective at removing BEs. And so, they should not be considered for
342 such purposes.

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

346 If the experiment is moderately unbalanced, and good precision is desired, Harman,
347 ComBat, Ratio-A, Ratio-G and BMC is a good choice with acceptable recall (While
348 Harman results in lower recall compare with others). If good recall is desired, then SVA
349 is a good choice with acceptable precision.

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

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

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

approaches help the analyst to evaluate the magnitude of a batch effect, 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 BEs at all, it does 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.

4. Materials and Methods

4.1 Simulated datasets

We simulate data modeled from the parameters of four real datasets: Three RNA-Seq datasets (genomics datasets), each dataset with three pairs of case and control samples (Ferreira et al., 2014; Isogai et al., 2018; Hatzi et al., 2019), and an unlabelled shotgun proteomic dataset with four cases and controls (Shao et al., 2010). The summary information of these datasets is shown in Table 1. An overview is provided in Fig. 5 linking datasets to the respective analytical output and corresponding figures as referenced in this manuscript.

We use the R package “polyester” to estimate the dispersion parameters of both RNA-Seq and proteomics samples, and fit these into a negative binomial distribution (Frazee et al., 2015). To demonstrate a good fit between simulated and real data, we use the mean/ standard deviation plots (Fig. S29).

We simulate 100 expression studies for genomics and proteomics datasets, respectively. For each study, we simulate expression for 1,000 genes or proteins on 20 samples with no class effects and batch effects. We use 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. S29).

In this study, we simulate three scenarios of confounding: (i) balanced batch-class ($R^2 = 0.0$), (ii) moderately unbalanced batch-class ($R^2 = 0.4$), (iii) 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 (Leek, 2014). Where R^2 is 0.8, it means there is 80% confounding between class and batch factors.

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

4.1.1 Method 1 (Non-uniformity assumptions)

For method 1, class effects and batch effects are incorporated based on the method of Leek (Leek, 2014). For each simulated study (genomics and proteomics), 20 samples are split into two classes (10 samples in each class, respectively). 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 BEs are simulated only on a subset of genes, this is referred to as a non-uniform approach.

4.1.2 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 (Langley and Mayr, 2015). Samples are randomly assigned to classes A and B (10 samples per class). For each class B sample in the simulated genomics/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 .

BEs 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 (Goh and Wong, 2017). Because a batch effect is inserted into every gene in batch 2 samples, this is referred to as a uniform approach.

4.2 Normalization methods

4.2.1 Linear scaling

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

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

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

4.2.2 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.

4.2.3 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.

4.3 Batch effect-correction algorithms (BECAs)

4.3.1 ComBat

ComBat is a widely used batch effect-correction algorithm. 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 (Johnson et al., 2007; Chen et al., 2011). The version used here is the implementation from the “sva” R package (Leek et al., 2012).

4.3.2 Harman

Harman is a method based on principal components analysis (PCA) (Oytam et al., 2016). Firstly, the adjust data was separated into its principal components (PCs). Harman scans each PC to identify variance correlated with BEs, followed by removing the BEs 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 BEs correction with Harman at threshold 95%. The version used here is the implementation from the “Harman” R package (Oytam et al., 2016).

4.3.3 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 (Leek and Storey, 2007; Leek, 2014), and may be removed, thereby preserving only class effects. The version used here is the implementation from the “sva” R package (Leek et al., 2012).

4.3.4 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 (Luo et al., 2010). The versions used here are the implementations from the “bapred” R package (Hornung et al., 2016).

4.3.5 Batch mean-centering (BMC)

BMC assume there are multiplicative systematic batch effects (Sims et al., 2008). 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 (Hornung et al., 2016).

4.4 Batch-effect detection methods

4.4.1 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 principal components (PCs) (Giuliani, 2017). 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.

4.4.2 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 (Reese et al., 2013). 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 .

4.5 Data pre-processing

Prior to normalization and/or batch correction, both RNA-Seq and proteomic measurements are log-transformed (base 2) to reduce scale and provide better data distribution symmetry (Leek, 2014).

4.6 Performance evaluation

4.6.1 Statistical feature selection analysis

Statistical feature selection is performed using the unpaired t-test with a P -value cutoff at 0.05. No multiple test corrections are performed, as these would artificially boost precision while penalizing recall (resulting in an unfair evaluation). Statistically significant genes/proteins are 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. Precision and recall are both 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:

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

4.6.2 Receiver operating characteristic (ROC) curves 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 generate 100 ROC curves per simulation. To summarize these, an averaged ROC curve is produced using the threshold-averaging approach as described by Fawcett (Fawcett, 2006) and used for overall performance evaluation. The ROC curves shown here are generated using the R packages “ROCR” (Sing et al., 2005) and “pROC” (Robin et al., 2011).

Acknowledgments

WWBG gratefully acknowledges Limsoon Wong, National University of Singapore, for inspiring this work. WWBG gratefully acknowledges support from the National Research Foundation of Singapore, NRF-NSFC (Grant No. NRF2018NRF-

NSFC003SB-006). The authors would also like to express their gratitude to the anonymous reviewers for their insights.

References

- Broadhurst, D., Goodacre, R., Reinke, S.N., Kuligowski, J., Wilson, I.D., Lewis, M.R., Dunn, W.B., 2018. Guidelines and considerations for the use of system suitability and quality control samples in mass spectrometry assays applied in untargeted clinical metabolomic studies. *Metabolomics*. 14, 72.
- Chen, C., Grennan, K., Badner, J., Zhang, D., Gershon, E., Jin, L., Liu, C., 2011. Removing Batch Effects in Analysis of Expression Microarray Data: An Evaluation of Six Batch Adjustment Methods. *Plos One*. 6, e17238.
- Fawcett, T., 2006. An introduction to ROC analysis. *Pattern Recognition Letters*. 27, 861-874.
- Ferreira, T., Wilson, S.R., Choi, Y.G., Risso, D., Dudoit, S., Speed, T.P., Ngai, J., 2014. Silencing of odorant receptor genes by G protein $\beta\gamma$ signaling ensures the expression of one odorant receptor per olfactory sensory neuron. *Neuron*. 81, 847-859.
- Frazee, A.C., Jaffe, A.E., Langmead, B., Leek, J.T., 2015. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics*. 31, 2778-2784.
- Giuliani, A., 2017. The application of principal component analysis to drug discovery and biomedical data. *Drug Discov Today*. 22, 1069-1076.
- Goh, W.W.B., Sng, J.C., Yee, J.Y., See, Y.M., Lee, T.S., Wong, L., Lee, J., 2017. Can Peripheral Blood-Derived Gene Expressions Characterize Individuals at Ultra-high Risk for Psychosis? *Comput Psychiatr*. 1, 168-183.
- Goh, W.W.B., Wang, W., Wong, L., 2017. Why Batch Effects Matter in Omics Data, and How to Avoid Them. *Trends in Biotechnology*. 35, 498-507.
- Goh, W.W.B., Wong, L., 2017. Protein complex-based analysis is resistant to the obfuscating consequences of batch effects --- a case study in clinical proteomics. *BMC Genomics*. 18, 142-142.
- Goh, W.W.B., Wong, L., 2018. Dealing with Confounders in Omics Analysis. *Trends in Biotechnology*. 36, 488-498.
- 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., 2019. Histone demethylase LSD1 is required for germinal center formation and BCL6-driven lymphomagenesis. *Nat Immunol*. 20, 86-96.
- Hornung, R., Boulesteix, A.-L., Causeur, D., 2016. Combining location-and-scale batch effect adjustment with data cleaning by latent factor adjustment. *BMC Bioinformatics*. 17, 27.
- Isogai, Y., Wu, Z., Love, M.I., Ahn, M.H., Bambah-Mukku, D., Hua, V., Farrell, K., Dulac, C., 2018. Multisensory Logic of Infant-Directed Aggression by Males. *Cell*. 175, 1827-1841.
- Jaffe, A.E., Hyde, T., Kleinman, J., Weinberg, D.R., Chenoweth, J.G., McKay, R.D., Leek, J.T., Colantuoni, C., 2015. Practical impacts of genomic data "cleaning" on biological discovery using surrogate variable analysis. *BMC Bioinformatics*. 16, 372.
- Johnson, W.E., Li, C., Rabinovic, A., 2007. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*. 8, 118-127.

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

Lazar, C., Meganck, S., Taminiau, J., Steenhoff, D., Coletta, A., Molter, C., Weiss-Solis, D.Y., Duque, R., Bersini, H., Nowe, A., 2012. Batch effect removal methods for microarray gene expression data integration: a survey. *Briefings in Bioinformatics*. 14, 469-490.

Leek, J.T., 2014. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic acids research*. 42, e161.

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

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

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

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., 2010. A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *Pharmacogenomics Journal*. 10, 278-291.

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

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

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., 2013. A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. *Bioinformatics*. 29, 2877-2883.

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

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

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., 2010. Shotgun proteomics analysis of hibernating arctic ground squirrels. *Mol Cell Proteomics*. 9, 313-326.

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

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

Table legends

Table 1. The datasets used in this study

Figure legends

Fig. 1. Batch-effect correction in genomics data 1 (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. Corresponding results for Genomics data 2 and 3 and Proteomics data simulations are shown in **Fig. S1 to S3**.

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

Fig. 3. Performance evaluation on precision/recall and ROC (Genomics data 1, 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. 4. The impact of prior normalization on BECAs (Genomics data 1, Non-uniformity assumption for batch-class, 100 simulations).

Fig. 5. Overview of simulation pipeline from dataset to results.