ELSEVIER

Feature

How missing value imputation is confounded with batch effects and what you can do about it

Wilson Wen Bin Goh [1,2,3,*], **Harvard Wai Hann Hui** [1,2], **Limsoon Wong** [4,5,*]

[1] Lee Kong Chian School of Medicine, Nanyang Technological University, Singapore
[2] School of Biological Sciences, Nanyang Technological University, Singapore
[3] Center for Biomedical Informatics, Nanyang Technological University, Singapore
[4] Department of Computer Science, National University of Singapore, Singapore
[5] Department of Pathology, National University of Singapore, Singapore

In data-processing pipelines, upstream steps can influence downstream processes because of their sequential nature. Among these data-processing steps, batch effect (BE) correction (BEC) and missing value imputation (MVI) are crucial for ensuring data suitability for advanced modeling and reducing the likelihood of false discoveries. Although BEC–MVI interactions are not well studied, they are ultimately interdependent. Batch sensitization can improve the quality of MVI. Conversely, accounting for missingness also improves proper BE estimation in BEC. Here, we discuss how BEC and MVI are interconnected and interdependent. We show how batch sensitization can improve any MVI and bring attention to the idea of BE-associated missing values (BEAMs). Finally, we discuss how batch-class imbalance problems can be mitigated by borrowing ideas from machine learning.

Keywords: class-batch proportion imbalance; batch effects; confounding; data science; computational biology; missing value imputation; statistics

Data processing steps are interconnected and interdependent

A typical functional analysis pipeline for high-throughput '-omics analysis requires a series of sequential steps, from raw data acquisition, to MVI, BEC, and normalization, before differential effect analysis (DEA) is performed.[1] The series of sequential steps also means that an upstream processing step would directly impact the downstream process. We have previously shown that the choice of normalization method influences BEC.[2] Other dependencies also exist but remain poorly explored and understood. Here, we discuss potential interactions between MVI and BE correction algorithms (BECAs), as well as the added complexities contributed by batch-size and batch-class imbalances and differing missing proportions between batches.

Missing values (MVs) in data are instances where a biological moiety is sporadically observed. MVs are usually taken care of by MVI methods[3,4] and can be categorized into three types: Missing Not At Random (MNAR); Missing Completely At Random (MCAR); and Missing At Random (MAR).[5] MNAR are usually associated with low-abundance biological moieties that occasionally fall below instrument detec-

1359-6446/
https://doi.org/10.1016/j.drudis.2023.103661

tion limits. MAR and MCAR are generally randomly distributed and are occasionally difficult to distinguish, particularly on smaller data sets. However, MAR depends on other attributes; that is, there are specific patterns in the data (or in auxiliary variables not in the data) that lead to specific patterns of missing values. By contrast, MCAR has no such dependency. The reasons for, and frequency of, MVs vary depending on the field of ‘-omics discussed. For example, in mass spectrometry-based untargeted metabolomics and proteomics data (~30–60% missingness),[6,7] MVs can occur because of deficiencies in peak detection or when compounds are co-eluted. These MV mechanisms would fall under the MAR/MCAR category. MNAR generally occurs in these types of data when the detection signal falls under the limit of detection. This type of missing data is challenging to interpret, because we cannot easily determine whether it stems from a biological issue, such as the true absence of the biological moiety, or a technical issue pertaining to low abundances or poor ionization. MVs in single cell RNA-sequencing (scRNA-seq) technologies (~30%)[8] are commonly termed as ‘dropouts’, and can stem from poor efficiency in gene capture, low gene expression, and stochastic gene expression across cells.[9]

Notably, MVs can occur across both features and samples. When MVs occur across a feature, it might be because of a technical issue, such as a limitation in sensitivity, or a biological issue in which the abundance of the target is too low, or the target is simply absent. By contrast, when MVs occur across a sample, it is likely because of technical issues specific to the sample or the batch in which it belongs to. This issue becomes more apparent when samples from several batches are integrated into a single data set.[10]

With regards to the extent of missing data, the technology at hand also matters. For example, unlike RNA-seq technologies, array-based gene expression data typically contain low levels of missing data (~1–10%). [11] This is because specific probes are designed to each hybridize with a predefined target sequence and, thus, the transcripts of different genes do not need to compete with those of other genes to be sampled and measured; by contrast, RNA-seq seeks to measure the expression levels of all genes present[12] and their transcripts have to compete to be sampled and measured. However, MVs remain prevalent in microarray data because of poor hybridization, scratches, and errors in the slides, among others.[13] Nevertheless, these ‘-omics sciences suffer from similar limitations regarding high-throughput technologies, and MVI and BECAs are both important steps in their data-processing pipelines.[14–16] Hence, dependencies between MVI and BECAs are relevant regardless of the ‘-omics in question.

MVI development is a complex and exciting area, where there is a wide variety of techniques addressing various data types and MV types.[17] Different techniques exist for resolving MNAR from MAR/MCAR MVs. Mixed-technique models capable of dealing with all three are rare. MVs are considered a serious issue, especially in high-throughput technologies with limited resolutions and high noise,[18] as mentioned above. Many powerful analysis methods downstream require completed data, which necessitates the use of MVIs. Of course, the proper selection of MVI is necessary for reducing false positives and false negatives.[17]

BEs are technical sources of variation that can arise from systematic differences between machines, experimenters, reagent lots, and so on.[14,19–21] They can cause both false positives and false negatives during DEA, and can be removed by BECAs. Examples of BECAs include ComBat and Combat-Seq,[15] Harman,[22] SVA,[23] and Batch-Mean Centering (BMC).[24] However, the proper application of BECAs is necessary, because wrong usage can generate errors.[15,25] Interestingly, most BECAs cannot deal with MVs and require a full data matrix as input. Therefore, MVI is usually performed upstream of BEC. There is a notable exception, HarmonizR,[26] which works by first combining data into one matrix. It then extracts submatrices that do not have too many missing values and independently batch corrects each submatrix based on user input, before stitching the corrected submatrices together again. This means that HarmonizR works around MVs by dissecting the matrix into regions without MVs. It then batch-normalizes the non-missing regions. This way, it aims to perform BEC without being affected by MVs. MVI can then be performed on the whole re-merged matrix as the last step. However, the creators of HarmonizR state that, when MVI is needed, it should preferably be performed before BEC rather than after.

At the BECA level, issues can also arise when the batch-class sizes are imbalanced. This refers to an imbalanced experimental design, such that, within a single batch, one class might have much fewer samples than another, or is entirely absent from the batch.[2,27] Avoiding such situations is not always possible, because of factors such as poor sample availability (e.g., obtaining samples for a rare disease) or failed experiments. The imbalance in batch-class proportions is akin to the *en masse* missingness of entire samples and can hinder the proper correction of BEs. It was shown in a previous study that, when the experiment design was severely unbalanced, BECA performance suffered and consequently affected downstream analysis outcomes.[2]

For most MVIs, batch information is not considered *a priori*. When imputation is performed across batches, this can lead to signal obfuscation, whereby the imputed value is essentially some artificial value averaged between the batches. Furthermore, the problem does not stop there: when the mis-imputed data matrix is subsequently fed to a BECA, this can cause under- and overcorrection, which, in turn, can impact DEA.

Despite being logically intuitive, MVI-BE confounding (MBC) is poorly characterized and there are few related studies. In both simulations and on real data benchmarks, it has been shown that MBC can create non-negligible issues on DEA.[28] The problem can be deceptively subtle: when MVI is misapplied in the presence of a BE, it can result in increased sample variances after BEC, creating issues with false positives and false negatives during DEA, while misleadingly appearing as though BEs have been eradicated.[10]

Given that MBC is a complex and newly characterized issue with direct effects on how much useful information we can get out of data, it is important to better understand the underlying issues: namely, how exactly MVIs can be led astray by BEs, how BECAs can be thwarted by MVs, and what can be done.

How do missing value imputation methods work (broadly) and how are MVIs misled by presence of batch effects?

Here, we provide a short introduction into how MVIs work. Broadly, MVIs are divisible into three categories: fixed (also known as naïve) imputation methods (e.g., zero or mean imputation), feature-based (e.g., the K-Nearest Neighbors; KNN), and multivariate (e.g., Multivariate imputation by chained equations; MICE).

Fixed imputation methods (FIMs) insert a static value for MVs belonging to the same biological moiety. Among FIMs, zero imputation is obviously wrong and can cause severe issues with downstream analysis.[29,30] This is because MVs can derive from the failure to detect the biological moiety, possibly because of low abundance. Given that this would not constitute a true zero, replacing the MV with zero will greatly underestimate the true value. Furthermore, if the proportion of MVs is high, zero imputation can also introduce false correlations between variables. Mean imputation is easy to understand and commonly used. MVs are filled in based on the average or median of non-missing observations given the same biological moiety. This approach disregards the presence of any batch factors in the data. It has previously been shown that, during mean imputation, ignoring batch information can be misleading.[28] In addition, mean imputation is often not conducted correctly. Some users forget that the mean or median should be derived from non-missing observations of the same biological moiety and, instead, incorrectly use the global mean or median over all biological moieties/class labels. FIMs also include minimum value imputations, such as the limit of detection (LOD) imputation, which replaces MVs with the minimum value of the sample.[3,31] This form of MVI is deemed to be simple and effective when handling left-censored data but fails to accurately retain class information, especially when MVs are MCAR.[3,32]

Feature-based methods use the concept of broad similarities calculated on observed variables to identify a set of samples most like the sample with the MV, allowing for 'informative' transfer of information only from those similar samples. A representative example is KNN, which is a classic machine learning approach used in both supervised and unsupervised contexts. In KNN, we identify neighboring points through a measure of distance and the MVs can be estimated using completed values of neighboring samples. There are various implementations of KNN for MVI (e.g., KNN-Mean and KNN-Imputer).[33] Similar to classic KNN for unsupervised learning, KNN as MVI also requires identification of an optimal set of *k* most similar samples. In many implementations, *k* is taken by default as the square root of *N*, the sample size. This default value of *k* does not consider any crucial structure in the data (including the presence of outliers), and certainly not the proportion of mixtures between batches found in the neighborhood. Deliberately increasing *k* increases the cross-batch mixture, but this likely simultaneously decreases DEA performance. Again, this informs us that batch sensitization of MVI is important, especially for methods the parameters of which can result in batch mixing.

Multivariate methods (MVMs) use various alternative features to converge on a 'best guess' for an MV. Among MVMs, Multiple Imputation by Chained Equations (MICE) is a popular method,[34] which works by multiple iterations of predictions. Placeholder values from complete cases are used to initiate the imputation process, and predicted values are fed back until stable.

Although it is common practice to remove features affected with MVs, it is recognized that doing so would result in the potential loss of important information. Generally, MVs in different '-omics studies are often handled differently, particularly in their imputation approaches. In mass spectrometry-based proteomics and metabolomics, Random Forest,[35] Bayesian PCA,[36] forms of KNN,[37] and mechanism aware methods[38] have been developed for MVI. Several MVI methods have also been developed specifically for scRNA-seq; some focus on preserving biological zeros, whereas others use data-driven machine learning approaches to recover MVs. By contrast, local-least squares-based approaches of MVI have been used for microarray data sets.[13,39]

There is also wisdom in refusing to impute when imputation is not necessary. A general recommendation is that if MVs account for <15% of data, you might be better off simply dropping those sample instances containing MVs (as opposed to dropping the features).[17] This case implicitly assumes that the MVs are MCAR; hence, dropping sample instances are akin to the effect of a random subsample. If the MVs are MAR or MNAR, they are dependent on either patterns of other data values in these sample instances or the actual values of the MVs themselves; thus, dropping these sample instances can lead to systematic biases that confound downstream analysis.[40] When MVs are >15% of the data, it would perhaps be more sensible to drop the MV-laden features, because there is little hope for MVI methods to work well on these MVs. There are also conflicting and extreme recommendations: while some hold that ~5% missingness produces negligible biases following MVI,[40] others have stated that 5% missingness should be the upper limit (for large data sets).[41] There has also been advice that, when imputing on 10% missingness, bias should be of concern and that, at 40% missingness, the imputed data should be viewed only as hypothesis generating.[42]

Notably, different MV proportion cutoffs are used in different '-omics studies. Even then, several approaches exist to determine them. In metabolomics, for example, one might follow the '80% rule',[30] which refers to retaining only features with at least 80% observations. However, such defined thresholds are often arbitrary and, thus, data-driven approaches have also been developed,[43] which make use of blank samples, MV proportions, and intraclass correlations to identify and drop features that are deemed uninformative. In proteomics, no specific guide on a missing threshold exists. Although we might turn to the 15% cutoff described above, it is still advisable to assess your data set for a more case-appropriate threshold. One method is to use OptiMissP,[44] an R dashboard that visualizes changes in the data with varying missing thresholds, allowing the user to make a more informed decision on a missing threshold that least alters the distribution of the data.

In some cases, features of importance might contain too many MVs and dropping them would lead to undesirable downstream analysis outcomes. For example, in a study involving participants who include both smokers and non-smokers, it is possible to observe the presence of nicotine derivatives, such as cotinine, in samples from smokers but not in those from non-smokers.[45] In such cases, the missingness in the data can be informative and a binary analysis that categorizes samples as either present or missing might be a possible approach. Alternatively, a zero-inflated method can be used to account for the excessive number of samples with zero levels of the compound in the non-smoking group. Thus, a purely statistical standpoint might not always be sufficient to decide whether to drop or impute a set of features. The decision to impute might also need to be driven by the importance of the specific features, how the imputation should be done, and whether the imputed values make sense.

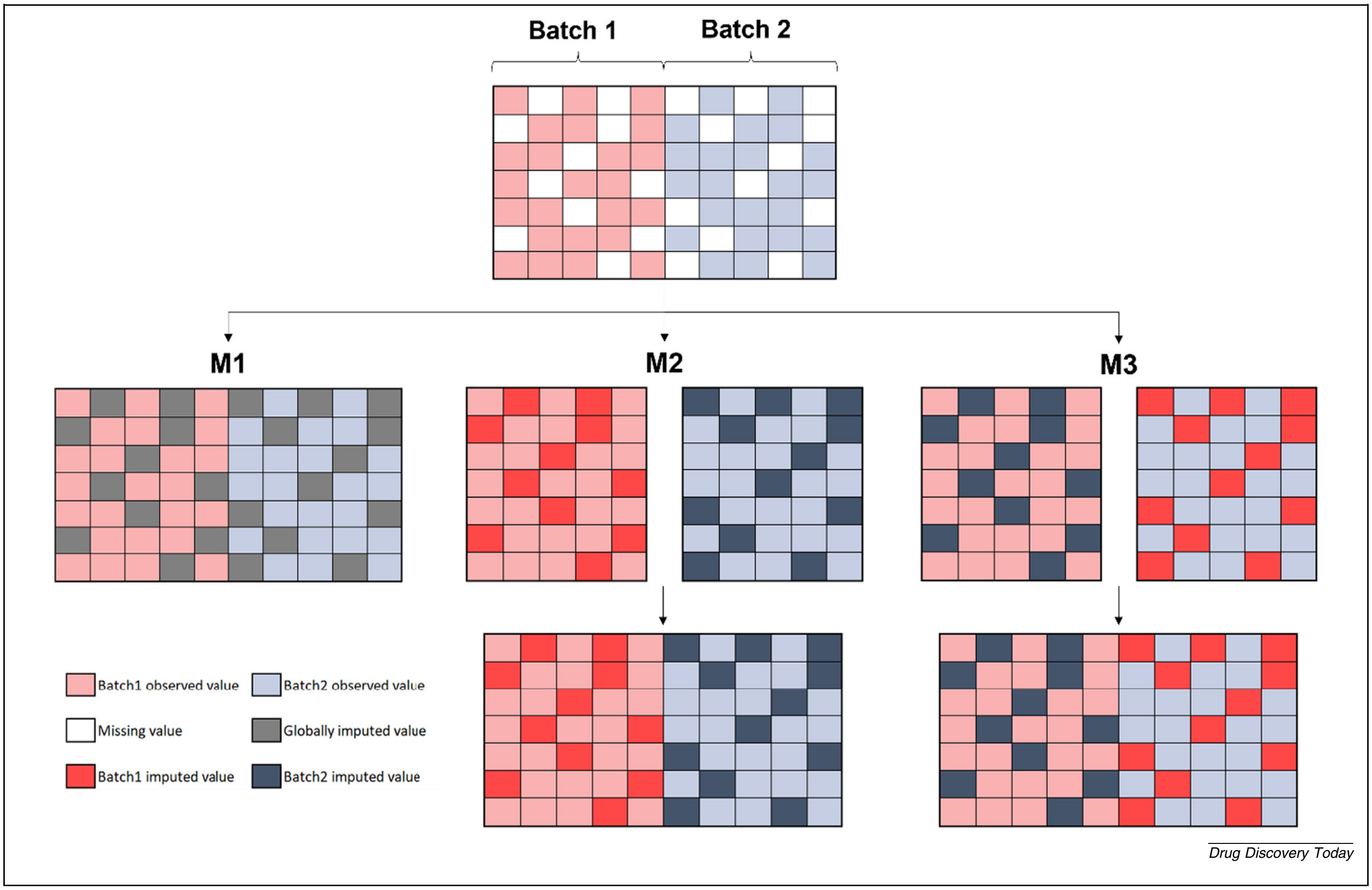
A key point to note is that MVI methods were never developed to explicitly consider batch factors. Yet, by their construction, many MVI strategies could be impaired by BEs. In the next section, we will take a deeper look into what happens during MBC using the example of mean imputation.

Batch-sensitizing MVI methods improves batch effect correction

We recently explored and evaluated how batch sensitization in MVI impacts BEC (which subsequently impairs the ability to identify correct gene targets) by modeling three simple imputation strategies [global (M1), self-batch (M2) and cross-batch (M3)] first via simulations, and then corroborated on real proteomics and genomics data (Figure 1).[28]

Developing a batch-sensitized approach is simple: we simply split the single-class/-moiety data up by batch and mean impute the MVs of each batch separately (we call this approach M2.) We compare M2 against a typical imputation strategy, which takes the global mean with no regard for the batch factor (we call this M1.) For a drastic contrast against M2, we purposely perform cross-batch imputation, such that the imputed mean value comes from the opposite batch (we call this M3).

We found that M1 and M3 impair all evaluated BECAs, including the popular method ComBat. This observation is platform independent, consistent on simulations, proteomics, and genomics data. We also found that the batch-sensitized approach, M2, is superior and consistently outperforms M1 and M3 on DEA. A crucial finding is that both M1 and M3 result in increased sample variance following BECA processing, whereas this is not the case for M2. For M2, the reduced sample variance is similar to the original batch-cleaned data (without MVs). This suggests that, when imputing missing values without regard for the batch factor, we end up exchanging BEs for noise, rendering them unrecognizable by BECAs and, therefore,



Drug Discovery Today

FIGURE 1

Strategies for batch sensitization and desensitization on missing value imputation (MVI). M1 is a typical non-batch-sensitized strategy where information is aggregated from all samples irrespective of the batch for imputation. M2 is batch sensitized, and only same batch samples are used for imputation. M3 is the opposite of M2, drawing information only from the opposite batch (this scenario is meant to exemplify the importance of batch sensitization, which might otherwise be hidden in M1).

uncorrectable. The increased sample variance has direct implications for DEA, resulting not only in false positives and negatives, but also mis-estimations of effect sizes.

Nonetheless, even with a good batch-sensitized strategy, we found that FIMs enhanced with M2 could never quite recover the original effect size (this is certainly a good argument as to why proper experiment design trumps post-hoc data salvage.) With increasing proportions of MVs, this problem gets worse. Although one could argue that feature-based methods and MVMs could do better, these methods are also constrained by their design: they require meaningful borrowable information from other samples and variables, respectively.

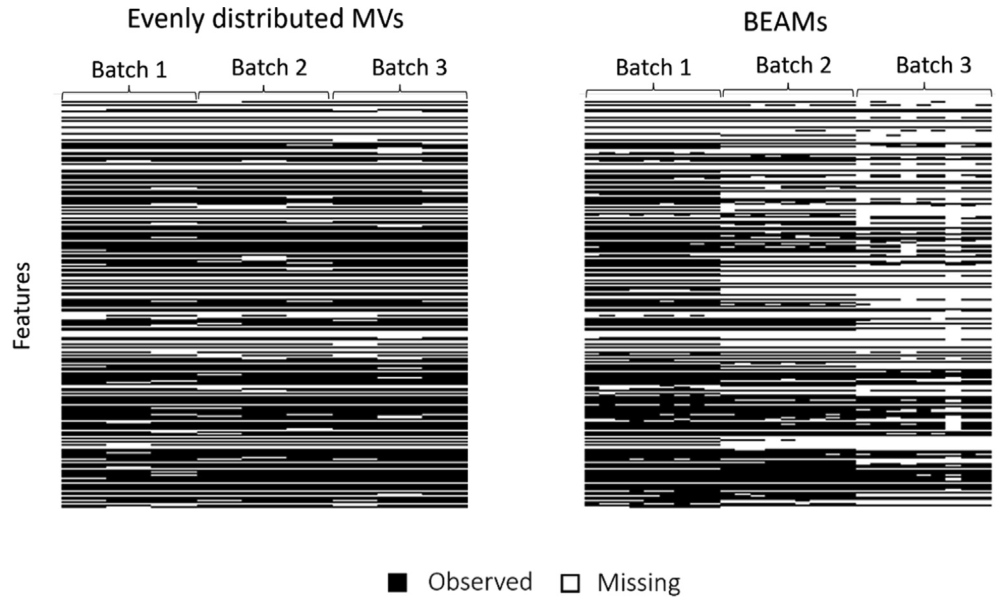
Thus, where both BEs and missing values are present in data, we recommend caution, and to ensure that the batch factor is taken into consideration early during the data-processing stage.

Batch effect-associated missing values create additional challenges for resolving batch effect-MVI confounding issues

We normally assume that MVs are evenly distributed across batches, such that batch sensitization [i.e., splitting by batch (M2)] is sufficient. However, we have found nothing instructing what to do with MVs that manifest as BEs. BEAMs are MVs associated with batches, such that, in one batch, there is a pronouncedly higher degree of missingness than in the other (Figure 2).[10] These MVs can be mixtures of MNAR, MAR, and MCAR. However, we think that BEAMs are likely mostly MAR because this depends on values in the data set. In this case, the values have BE components.

When MVs are batch associated, they should be mostly specific to the batch they are in. This suggests that batch sensitization (M2) should still work well. When batch-associated MVs are usually MAR, there should be other variables in the data (or auxiliary variables not in the data), which are highly correlated with the MV-laden variables. Then, MVI might still perform reasonably well despite large amounts of missingness,[41] provided all such variables highly correlated with the MV-laden variables are made available in the data. Nonetheless, we can also intuitively infer that MVI performance is likely inversely correlated to the amount of missingness: the greater the number of MVs in the samples, the higher the likelihood of having missing values in correlated variables as well. As a result, there is less information available to accurately perform MVI.

Hence, when there are several batches, such that some batches have many more MVs, a more elaborate strategy than M2 might be required. One possible idea is to perform a stepwise imputation (SIM) so that MV-laden batches can borrow information from batches with fewer MVs. Although this SIM approach has not been proven in the context of BEs, it is in alignment with the default 'impute feature with fewest missing value first' option of IterativeImputer, which is the main MVI provided by the very popular scikit-learn Python software package. In SIM, MVI is performed sequentially on the batches from least to most missingness. Following each MVI step, the imputed values are then transferred to the next batch in order of increasing missingness. SIM can be tricky to get right because it has similar problems to M3 if the imputed values are transferred directly without attempting to perform some form of batch adjustment *a priori*. For example, a batch correction factor could be estimated based on observed components across batches. This information could then be used to adjust the imputed values on the confident batch



Drug Discovery Today

FIGURE 2

Batch effect-associated missing values (BEAMs). In **(a)**, the distribution of missingness is even across each batch. In **(b)**, the batches are associated with greater missingness from left to right. This imbalance of missingness that is batch specific is referred to as BEAMs and can have implications for missing value (MV) imputation.

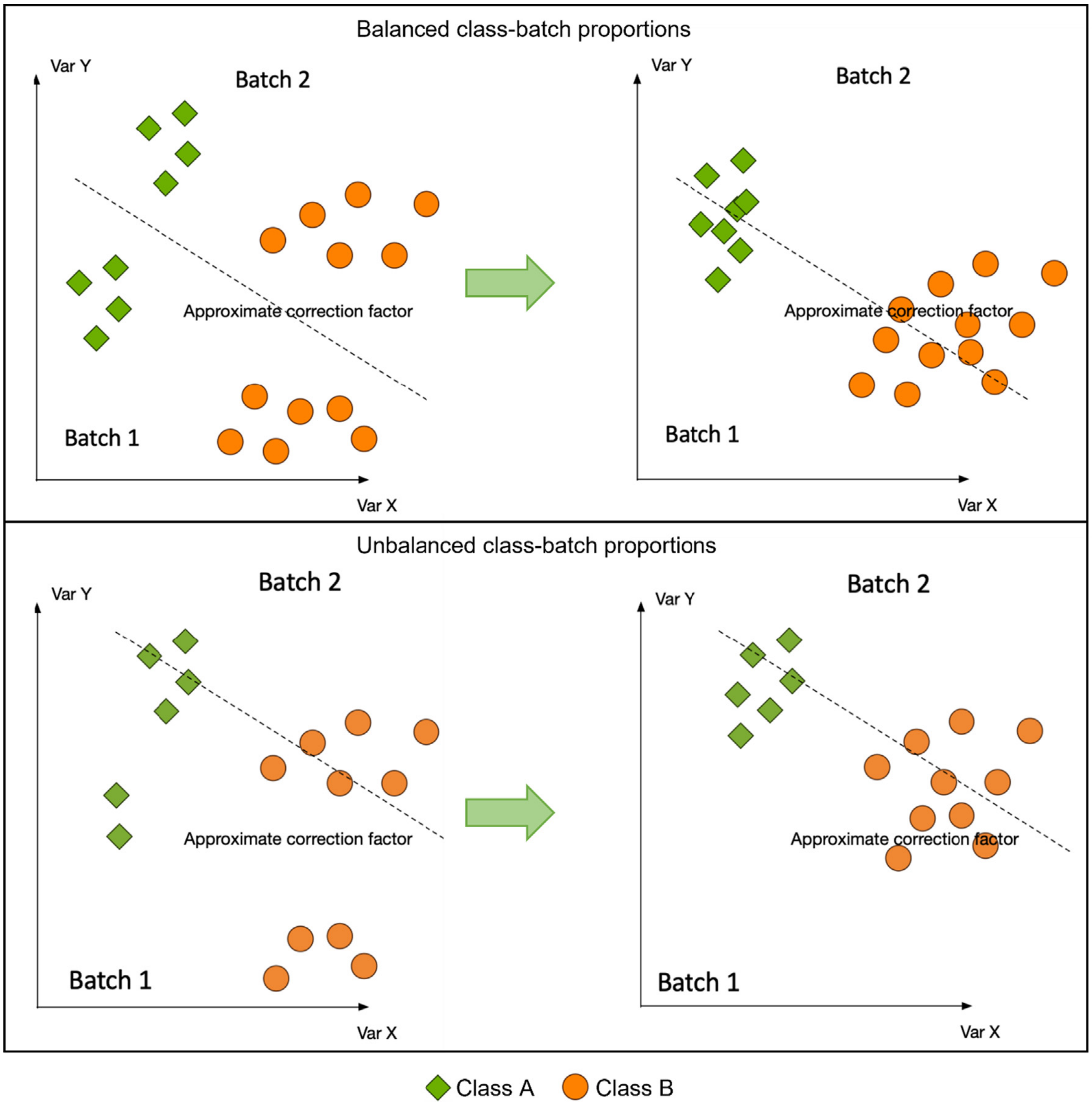
with lower missingness before transfer to the next batch with higher missingness.

It is difficult to say when BEAM issues are pertinent enough to warrant alternative strategies, such as SIM over M2. We encountered the issue when we found that, given the same sample, technical replicates across machines generated missingness rates of 10%, 30%, and 50%, respectively (data from the Clinical Proteomic Tumor Analysis Consortium; CPTAC).[46] In such situations, we showed that a pure M2 batch-sensitization approach is insufficient because there is a missingness hierarchy across batches, which causes an overall loss of coverage limited to the batch with lowest missingness. A global M1 would be even less appropriate, because the imputation result would be driven essentially by the batch with least missingness.[10]

Batch-class imbalance can impede proper batch effect correction

So far, we have taken an MVI-centric perspective. That is, we considered how various strategies of MVI impact BEC. This is primarily because MVI is usually performed upstream of BEC. However, there is another layer to consider, specifically in terms of how BECAs work.

Most BECAs make naïve assumptions on data. A typical assumption for optimal application of ComBat is that there is no batch-size and batch-class imbalance in data (Figure 3). Whereas, when there is batch-size imbalance or even worse, batch-class imbalance, the eventual correction efficacy is affected, resulting in mis-


Balanced class-batch proportions
Var Y
Batch 2
Approximate correction factor
Batch 1
Var X
Var Y
Batch 2
Approximate correction factor
Batch 1
Var X
Unbalanced class-batch proportions
Var Y
Batch 2
Approximate correction factor
Batch 1
Var X
Var Y
Batch 2
Approximate correction factor
Batch 1
Var X
Class A
Class B

Drug Discovery Today

FIGURE 3

How class-batch proportion imbalance can affect batch effect correction. Class labels are shown as different shapes. The batch effect is shown as a separation of the shapes across the diagonal (Batch 1 is below and Batch 2 is above). In the top row, balanced class-batch proportions result in an unbiased batch effect correction, In the lower row, because there are more samples in batch 2, it will bias the correction factor towards itself.

correction of BEs on data, with consequences for DEA[2] and other downstream analysis. In such situations, this is akin to the *en masse* missingness of entire samples (of the minority classes or the small batches, as the case might be) that could have given us a better estimate of class and batch representation. A crucial example was pointed out by Yong *et al.*[47] in the context of scRNA-seq analysis: when there are some rare mutant cell types (e.g., premalignant mutants and emerging treatment-resistant mutants) that are present only in some batches and not in other samples/batches, many BECAs specialized for scRNA-seq (in particular, Seurat,[48] LIGER,[49] and Harmony[50]) have been shown to miscorrect these mutant cells into their parental cell types; this loss of information would cause the nondetection of these mutant cell types.

Such problems with imbalance are well known in machine learning. High class representation disparities can lead to 'lazy learners', where a trained classifier achieves good performance accuracy simply by assigning all test samples to the majority class. Such imbalance problems (but not an entire class missing en masse) are typically handled by simple techniques, such as oversampling, undersampling, and Synthetic Minority Oversampling Technique (SMOTE).[27]

Oversampling replicates samples from the minority class in the training data set. Given that it creates many repeated instances that might not reflect the true data distributions, it might cause overfitting in trained models. By contrast, undersampling randomly selects samples from the majority class. Depending on whether the sampling is representative, it can result in losing important information in the model. A third alternative is SMOTE, which involves synthesizing new instances from the minority class. SMOTE works by selecting proximal samples in feature space, drawing a line between the samples, and generating a new sample at a point along that line. SMOTE can be powerful if its running parameters can be constrained to produce simulated data preserving the characteristics of the original data.[51]

All three techniques can be applied to address batch-size and batch-class imbalance issues, which impact BECAs. In a recent study, oversampling was performed to improve BEC before investigation for Doppelgänger effects in machine learning.[52] As for the observation of Yong *et al.* on rare cell types in the context of scRNA-seq BE correction and batch integration, it is more a biological than a technological phenomenon. Thus, there is not an MVI issue; instead, one should deploy a highly conservative BE correction to avoid losing the rare cell types.[19,47]

Recommendations

Given that there are many kinds of MVIs and MVs, identifying the right tool for performing MVI is important. Consider using this guide by Kong *et al.*[17] to help you decide an appropriate imputation strategy. More importantly, MVI is batch sensitive; thus, you should check whether BEs might be present in your data. One simple approach is to check the meta-information, but you can also inspect via visualizations or calculations based on common non-missing variables.

- When missingness is evenly distributed across batches, consider using a batch-sensitized M2 strategy for MVI.
- When missingness is non-evenly distributed across batches, unconventional strategies for MVI, such as stepwise imputation (SIM), might be necessary, but requires further investigation and development. Alternatively, a method such as HarmonizR can also be useful.[26].
- BECAs generally require balance in terms of class and batch sizes. An imbalance in this aspect can be regarded as a form of en masse missingness. Some methods used in machine learning can be borrowed to address this issue.
- Relevant approaches include oversampling, undersampling, and SMOTE. There does not appear to be consensus on which approach is optimal and it remains an open area for further research and understanding.
- Finally, when batches are unavoidable, adequate planning/designing of the batches is essential: batches should be roughly the same size, have roughly the same proportions of class types, and so on.

Concluding remarks

BEC is closely interdependent with MVI. Although identifying the correct MVI is important, such methods do need to take BEs into account to work optimally (i.e., batch sensitization). Where MVs are also batch associated, special care needs to be taken during MVI. An *en masse* form of missingness resulting in class-batch proportion imbalances also prevents BECAs from working optimally; these are addressable by borrowing data-balancing methods from machine learning. Finally, when batches are unavoidable, the planning/designing of batches must be performed sufficiently.

Authors' contributions

WWBG and LW cowrote the manuscript and supervised. HHWH developed the figures and contributed to editing.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research/project is supported by the National Research Foundation, Singapore under its Industry Alignment Fund – Pre-positioning (IAF-PP) Funding Initiative and a Ministry of Education (MOE), Singapore Tier 1 grant (Grant No. RG35/20). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of these funding agencies.

Declaration of interests

The authors declare no conflicting interests, financial or otherwise.

References

1. Goh WWB, Wong L. The birth of bio-data science: trends, expectations, and applications. *Genom Proteom Bioinf*. 2020;18:5–15.
2. Zhou L, Chi-Hau Sue A, Bin Goh WW. Examining the practical limits of batch effect-correction algorithms: when should you care about batch effects? *J Genet Genomics*. 2019;46:433–443.
3. Liu M, Dongre A. Proper imputation of missing values in proteomics datasets for differential expression analysis. *Brief Bioinform*. 2021;22: bbaa112.
4. Lin WC, Tsai CF. Missing value imputation: a review and analysis of the literature (2006–2017). *Artif Intell Rev*. 2020;53:1487–1509.
5. Rubin DB. Inference and missing data. *Biometrika*. 1976;63:581–592.
6. Hrydziuszko O, Viant MR. Missing values in mass spectrometry based metabolomics: an undervalued step in the data processing pipeline. *Metabolomics*. 2012;8:161–174.
7. Webb-Robertson BJM et al. Review, evaluation, and discussion of the challenges of missing

value imputation for mass spectrometry-based label-free global proteomics. *J Proteome Res*. 2015;14:1993–2001.

8. Yang MQ, Weissman SM, Yang W, Zhang J, Canaann R, Guan R. MISC: missing imputation for single-cell RNA sequencing data. *BMC Syst Biol*. 2018;12:114.
9. Lun ATL, Bach K, Marioni JC. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biol*. 2016;17:75.
10. Hui HWH, Goh WWB. Uncovering the consequences of batch effect associated missing values in omics data analysis. *bioRxiv*. 2023. https://doi.org/10.1101/2023.01.30.526187. Published online February 12, 2023.
11. de Brevern AG, Hazout S, Malpertuy A. Influence of microarrays experiments missing values on the stability of gene groups by hierarchical clustering. *BMC Bioinf*. 2004;5:114.
12. Liu H, Bebu I, Li X. Microarray probes and probe sets. *Front Biosci (Elite Ed)*. 2010;2:325–338.
13. Chiu CC, Chan SY, Wang CC, Wu WS. Missing value imputation for microarray data: a comprehensive comparison study and a web tool. *BMC Syst Biol*. 2013;7:S12.
14. Goh WWB, Wang W, Wong L. Why batch effects matter in omics data, and how to avoid them. *Trends Biotechnol*. 2017;35:498–507.
15. Zhang Y, Parmigiani G, Johnson WE. ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genom Bioinf*. 2020;2:lqaa078.
16. Han W, Li L. Evaluating and minimizing batch effects in metabolomics. *Mass Spectrom Rev*. 2022;41:421–442.
17. Kong W, Hui HWH, Peng H, Goh WWB. Dealing with missing values in proteomics data. *Proteomics*. 2022;22:2200092.
18. Zhou L, Wong L, Goh WWB. Understanding missing proteins: a functional perspective. *Drug Discov Today*. 2018;23:644–651.
19. Goh WWB, Yong CH, Wong L. Are batch effects still relevant in the age of big data? *Trends Biotechnol*. 2022;40:1029–1040.
20. Čuklina J et al. Diagnostics and correction of batch effects in large-scale proteomic studies: a tutorial. *Mol Syst Biol*. 2021;17.
21. Wang Y, LêCao KA. Managing batch effects in microbiome data. *Brief Bioinform*. 2020;21:1954–1970.
22. Oytam Y, Sobhanmanesh F, Duesing K, Bowden JC, Osmond-McLeod M, Ross J. Risk-conscious correction of batch effects: maximising information extraction from high-throughput genomic datasets. *BMC Bioinf*. 2016;17:332.
23. Leek JT. svaseq: removing batch effects and other unwanted noise from sequencing data. *Nucleic Acids Res*. 2014;42.
24. Nygaard V, Rødland EA, Hovig E. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics*. 2016;17:29–39.
25. Li T, Zhang Y, Patil P, Johnson WE. Overcoming the impacts of two-step batch effect correction on gene expression estimation and inference. *Biostatistics*. 2021. https://doi.org/10.1093/biostatistics/kxab039. Published online December 10, 2021.
26. Voß H et al. HarmonizR enables data harmonization across independent proteomic datasets with appropriate handling of missing values. *Nat Commun*. 2022;13:3523.
27. Kim M, Hwang KB. An empirical evaluation of sampling methods for the classification of imbalanced data. *PLoS One*. 2022;17.
28. Hui HWH, Kong W, Peng H, Goh WWB. The importance of batch sensitization in missing value imputation. *Sci Rep*. 2023;13:3003.
29. Knol MJ et al. Unpredictable bias when using the missing indicator method or complete case analysis for missing confounder values: an empirical example. *J Clin Epidemiol*. 2010;63:728–736.
30. Wei R et al. Missing value imputation approach for mass spectrometry-based metabolomics data. *Sci Rep*. 2018;8:663.
31. Hornung RW, Reed LD. Estimation of average concentration in the presence of nondetectable values. *Appl Occup Environ Hyg*. 1990;5:46–51.
32. Song M et al. A review of integrative imputation for multi-omics datasets. *Front Genet*. 2020;11.
33. Beretta L, Santaniello A. Nearest neighbor imputation algorithms: a critical evaluation. *BMC Med Inform Decis Mak*. 2016;16:74.
34. Azur MJ, Stuart EA, Frangakis C, Leaf PJ. Multiple imputation by chained equations: what is it and how does it work? *Int J Methods Psychiatr Res*. 2011;20:40–49.
35. Kokla M, Virtanen J, Kolehmainen M, Paananen J, Hanhineva K. Random forest-based imputation outperforms other methods for imputing LC-MS metabolomics data: a comparative study. *BMC Bioinf*. 2019;20:492.
36. Jin L et al. A comparative study of evaluating missing value imputation methods in label-free proteomics. *Sci Rep*. 2021;11:1–11.
37. Lee JY, Styczynski MP. NS-kNN: A modified k-nearest neighbors approach for imputing metabolomics data. *Metabolomics*. 2018;14:153.
38. Dekermanjian JP, Shaddox E, Nandy D, Ghosh D, Kechris K. Mechanism-aware imputation: a two-step approach in handling missing values in metabolomics. *BMC Bioinf*. 2022;23:179.
39. Dubey A, Rasool A. Efficient technique of microarray missing data imputation using clustering and weighted nearest neighbour. *Sci Rep*. 2021;11:24297.
40. Schafer JL. Multiple imputation: a primer. *Stat Methods Med Res*. 1999;8:3–15.
41. Madley-Dowd P, Hughes R, Tilling K, Heron J. The proportion of missing data should not be used to guide decisions on multiple imputation. *J Clin Epidemiol*. 2019;110:63–73.
42. Jakobsen JC, Gluud C, Wetterslev J, Winkel P. When and how should multiple imputation be used for handling missing data in randomised clinical trials – a practical guide with flowcharts. *BMC Med Res Methodol*. 2017;17:162.
43. Schiffman C et al. Filtering procedures for untargeted LC-MS metabolomics data. *BMC Bioinf*. 2019;20:334.
44. Arioli A et al. OptiMissP: a dashboard to assess missingness in proteomic data–independent acquisition mass spectrometry. *PLoS One*. 2021;16.
45. Avila-Tang E et al. Assessing secondhand smoke using biological markers. *Tob Control*. 2013;22:164–171.
46. Rudnick PA et al. A description of the Clinical Proteomic Tumor Analysis Consortium (CPTAC) common data analysis pipeline. *J Proteome Res*. 2016;15:1023–1032.
47. Yong CH et al. Mapbatch: conservative batch normalization for single cell RNA-sequencing data enables discovery of rare cell populations in a multiple myeloma cohort. *Blood*. 2021;138:2954.
48. Hao Y et al. Integrated analysis of multimodal single-cell data. *Cell*. 2021;184:3573–3587.e29.
49. Welch JD, Kozareva V, Ferreira A, Vanderburg C, Martin C, Macosko EZ. Single-cell multi-omic integration compares and contrasts features of brain cell identity. *Cell*. 2019;177:1873–1887.
50. Korsunsky I et al. Fast, sensitive, and accurate integration of single cell data with Harmony. *Nat Methods*. 2019;16:1289–1296.
51. Wang S, Dai Y, Shen J, Xuan J. Research on expansion and classification of imbalanced data based on SMOTE algorithm. *Sci Rep*. 2021;11:24039.
52. Wang LR, Choy XY, Goh WWB. Doppelgänger spotting in biomedical gene expression data. *iScience*. 2022;25.

Wilson Wen Bin Goh [1,2,3,*], Harvard Wai Hann Hui [1,2], Limsoon Wong [4,5,*]

[1] Lee Kong Chian School of Medicine, Nanyang Technological University, Singapore
[2] School of Biological Sciences, Nanyang Technological University, Singapore
[3] Center for Biomedical Informatics, Nanyang Technological University, Singapore
[4] Department of Computer Science, National University of Singapore, Singapore
[5] Department of Pathology, National University of Singapore, Singapore

* *Corresponding authors:*