OXFORD

Briefings in Bioinformatics, 2025, **26(2)**, bbaf168
https://doi.org/10.1093/bib/bbaf168
Case Study

Assessing the impact of batch effect associated missing values on downstream analysis in high-throughput biomedical data

Harvard Wai Hann Hui[1,‡], Wei Xin Chan[1,2], Wilson Wen Bin Goh[1,2,3,4,5,*]

[1]Lee Kong Chian School of Medicine, Nanyang Technological University, 59 Nanyang Drive, Singapore 636921, Singapore
[2]Center for Biomedical Informatics, Nanyang Technological University, 59 Nanyang Drive, Singapore 636921, Singapore
[3]School of Biological Sciences, Nanyang Technological University, 60 Nanyang Drive, Singapore 637551, Singapore
[4]Center for Artificial Intelligence in Medicine, Nanyang Technological University, 59 Nanyang Drive, Singapore 636921, Singapore
[5]Division of Neurology, Department of Brain Sciences, Faculty of Medicine, Imperial College London, Burlington Danes, The Hammersmith Hospital, Du Cane Road, London W12 0NN, United Kingdom

*Corresponding author. Lee Kong Chian School of Medicine, Nanyang Technological University, 59 Nanyang Drive, Singapore 636921, Singapore.
E-mail: wilsongoh@ntu.edu.sg
‡First author.

Abstract

Batch effect associated missing values (BEAMs) are batch-wide missingness induced from the integration of data with different coverage of biomedical features. BEAMs can present substantial challenges in data analysis. This study investigates how BEAMs impact missing value imputation (MVI) and batch effect (BE) correction algorithms (BECAs). Through simulations and analyses of real-world datasets including the Clinical Proteomic Tumour Analysis Consortium (CPTAC), we evaluated six MVI methods: K-nearest neighbors (KNN), Mean, MinProb, Singular Value Decomposition (SVD), Multivariate Imputation by Chained Equations (MICE), and Random Forest (RF), with ComBat and limma as the BECAs. We demonstrated that BEAMs strongly affect MVI performance, resulting in inaccurate imputed values, inflated significant *P*-values, and compromised BE correction. KNN, SVD, and RF were particularly prone to propagating random signals, resulting in false statistical confidence. While imputation with Mean and MinProb were less detrimental, artifacts were nonetheless introduced. Furthermore, the detrimental effect of BEAMs increased in parallel with its severity in the data. Our findings highlight the necessity of comprehensive assessments and tailored strategies to handle BEAMs in multi-batch datasets to ensure reliable data analysis and interpretation. Future work should investigate more advanced simulations and a variety of dedicated MVI methods to robustly address BEAMs.

Keywords: batch effects; biomedical informatics; genomics; missing values; proteomics; statistics

Introduction

Advanced technologies in proteomics, genomics, metabolomics, and transcriptomics are critical for biomarker development and drug target identification, enabling efficient, low-cost biomarker screening. However, their effectiveness is limited by small sample sizes, which results in high-dimensional data with more features than samples—a phenomenon known as the curse of dimensionality. To address this, data is often integrated from multiple sources to increase sample size. However, doing so almost always introduces batch effects (BE) into the data.

BEs are data biases arising from experiment design, processing, or collection methods. For instance, profiling the same cohort on two machines may result in distinct subgroups due to machine-specific BEs, which may lead to false positives/negatives in data analysis [1, 2]. BE correction (BEC) is thus necessary [2], with tools like ComBat [3], limma [4], Surrogate Variable Analysis [5], Remove Unwanted Variation [6], and Harman [7] widely used in proteomics and genomics. Other methods are platform-specific, such as those specialized for processing single-cell RNA-sequencing (scRNA-seq) data.

Missing values (MVs) present another important problem in high-dimensional biomedical data. These typically refer to features that were not observed in a sample due to either biological or technical reasons [8]. Biological MVs are usually related to the heterogeneity of biological systems. Technical MVs may arise from various reasons, such as machine limitations or sample loss during preparation. MVs are categorizable into three groups—Missing Not at Random (MNAR), Missing Completely at Random (MCAR), and Missing at Random (MAR) [8]. MNAR refers to missingness that depends on the value of the data point itself, such as when the abundance of a certain protein is below the limit of detection of the machine. MCAR refers to missingness that is completely random and is unrelated to other data points. Finally, MAR suggests that the probability of the data point being an MV depends on the observed data.

Depending on the goal, MVs are handled by (i) omitting MV-riddled features, (ii) using missing value imputation (MVI) methods to estimate them, or (iii) using statistical methods that can handle (or side-step) MVs [9]. To meet the requirements of many popular machine learning or statistical approaches such as principal component analysis (PCA) [10], obtaining a complete

Received: November 1, 2024. **Revised:** March 10, 2025. **Accepted:** March 24, 2025

dataset via MVI is often necessary, although such approaches can also introduce bias [11, 12]. Given the wide variety of MVI methods available, it is difficult to determine the most appropriate MVI technique that suits a particular data characteristic [8]. Furthermore, avoiding MVI entirely is often impossible as simply omitting features with MVs may result in a large loss of features and thus potentially useful information.

In a conventional analysis workflow, MVI usually precedes BEC, suggesting that imputation can impact batch correction. However, interactions between MVI and BEC remain understudied, despite evidence that they do confound each other [13]. Furthermore, MVs may manifest in a batch-specific manner, creating challenging scenarios similar to perfect class-batch confounding [14]. Such problems involving confounding between MVI and BEC may warrant dedicated approaches.

Batch effect associated missing values (BEAMs) refers to batch-wide MVs that arise from the integration of batches with different proteome (or genome) coverage. In previous studies, BEAMs were observed when integrating batches increased MV proportions because features were not detected in certain batches [15, 16]. Similarly, in the Clinical Proteomic Tumour Analysis Consortium (CPTAC) study 6 proteomics data, a widely accessed data resource, integration of four batches led to batch-specific MVs due to inconsistent coverage [17]. Similarly, MVs in scRNA-seq, also known as dropouts, have been shown to be dependent on batch conditions and varying detection rates [18, 19].

Some studies ignore batch structures (and BEAMs) by performing direct imputation on the CPTAC data [9, 20], which is far from ideal. Our previous work on cross-batch MVI (estimating MVs using a different batch), demonstrated that improper imputation in multi-batch data produces false effects, and that batch structures have to be properly handled during analysis [13]. BEAMs present a related but more challenging issue: In our cross-batch imputation study, blindly imputing features observed across all batches (i.e. without BEAMs) may induce batch mixing within an imputed feature. Imputing BEAMs, however, may result in a feature of a particular batch taking on values from another batch entirely.

Few MVI studies have addressed BEAMs in multi-batch data [21–23]. While these studies simulate MVs in varied ways, they overlook batch-associated missingness, making their findings inapplicable to BEAMs. In our previous work, we recommended a batch-sensitized MVI strategy for multi-batch data [13]. This approach is unsuitable for dealing with BEAMs due to the lack of within-batch observations for estimation. A unified MVI and BEC strategy based on causal inference was proposed to deal with BEAMs in scRNA-seq [24]. However, its current design restricts its applicability beyond scRNA-seq datasets. Given the limited literature on BEAMs, strengthening our understanding on the impact of imputing BEAMs using conventional methods on widely used bulk omics platforms is important for developing appropriate strategies.

In this paper, we study the impact of BEAMs on analysis involving six widely used MVI methods [K-nearest neighbors (KNN), Mean, MinProb, Singular Value Decomposition (SVD), Multivariate Imputation by Chained Equations (MICE), and Random Forest (RF)], in real-world and simulated datasets.

Methods

Study design

An overview of our study design for evaluating the impact of BEAMs on downstream analysis is provided in Fig. 1. The workflow consists of two parts: the artificial MV workflow (solid arrows) and the natural MV workflow (dotted arrows). The artificial MV workflow involves inserting MVs into the starting matrix, which can either be a simulated matrix or a complete real-world dataset (obtained by removing MV-laden features). MVs are inserted as either non-BEAMs (Control) or BEAMs, resulting in two distinct MV-laden datasets. These are then pre-processed by class-specific quantile normalization [25], MVI, and BEC. The evaluation of these datasets focuses on the BEs, imputation accuracy, inter-sample correlations, and differential expression analysis (DEA). On the other hand, the natural MV workflow uses the CPTAC dataset exclusively, following the same pre-processing steps as the artificial MV workflow. However, its evaluation is limited to assessing BEs and inter-sample correlations.

Datasets

Clinical Proteomic Tumour Analysis Consortium study six datasets

To portray BEAMs, we used the CPTAC study 6 shotgun proteomics dataset, which involves yeast samples spiked with Sigma UPS1 proteins at various concentrations [17]. Here, data from four locations served as the batches and were integrated into a single dataset. Each batch contained five spike-in concentration levels with three samples per level. In total, there were 60 samples.

Quartet proteomics and metabolomics datasets

Five batches and two classes from each of the Quartet proteomics [26, 27] and the metabolomics [28] datasets were used for additional analysis. These are small datasets, with each class-batch containing only one sample. The datasets were obtained from Yu *et al.* [29], with detailed information described in the respective papers of each dataset.

Data simulation

The datasets simulated in our study contain four batches and two classes with 5 samples in each class-batch, amounting to a total of 40 samples. We simulated 1000 features from a gamma distribution of log-expression values, with 20% of these containing class effects (see Supplementary Information for more details). The results in this study shown for the simulated datasets were derived from 10 rounds of simulations, unless otherwise stated.

Missing value simulation

For each real-world and simulated dataset, we simulated two types of MVs: (i) BEAMs and (ii) MVs not associated with BEs, with the latter serving as a negative control (Fig. 2).

To simulate MVs unrelated to BEs, we applied the method proposed by Jin *et al.*, which models MVs in proteomics data using both MCAR and MNAR mechanisms [23]. This method is parameterized by α, the total percentage of MVs, and β, the proportion of MNAR values. First, features with average expression values below the α-th quantile are identified. Then, a Bernoulli trial with probability β determines whether each identified feature value is dropped. Finally, MCAR is introduced at random to reach the total MV percentage α.

To simulate BEAMs, we adapted this approach by applying dropouts to entire batches instead of individual samples. Specifically, if the average expression of a feature within a batch falls below the α-th quantile, a Bernoulli trial with probability β determines whether all feature values in that batch are dropped. Based on observations of real-world proteomics datasets where MVs commonly range between 10% and 50% [22, 30, 31], we set $\alpha = 0.4$ to produce ~40% MVs. Since real-world MVs are predominantly MNAR, we set $\beta = 0.8$ to achieve a ~4:1 MNAR to MCAR ratio. Figure 3 shows that our simulated MVs closely

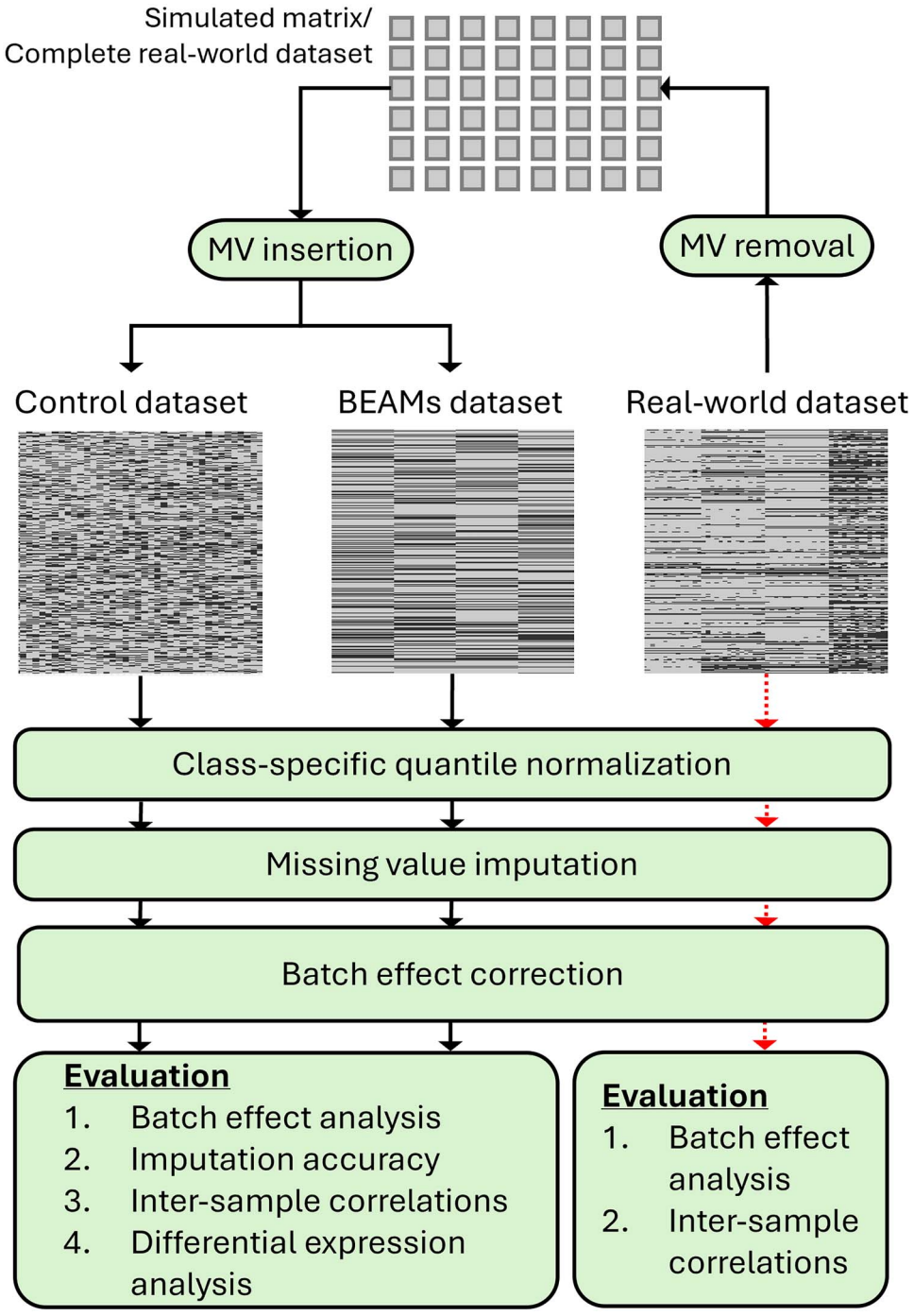

Figure 1. Overview of study workflow. Solid arrows indicate the workflow involving artificial MVs, while dotted arrows indicate the workflow involving natural MVs. The artificial MV workflow (solid arrows) begins with either a simulated matrix or real-world dataset with MVs removed. MVs are inserted as either non-BEAMs (control) or BEAMs. All MV-laden datasets are then normalized, before MVI and subsequent BEC. Evaluation was then performed based on the batch effect, imputation accuracy, inter-sample correlations, and DEA. The natural MV workflow (dotted arrows) involves only the CPTAC dataset and is evaluated by the batch effect and inter-sample correlations.

resemble those observed in real-world datasets whereby missing proportion decreases as the average feature expression increases.

Missing value imputation

We compared six commonly used MVI techniques: KNN, Mean, MinProb, SVD, MICE, and RF. For a broader perspective on MVI performance on BEAMs-laden data, the six methods selected each cover different operating principles such as local similarity, simple-substitution, left-censored, global-structure imputation, multiple imputation, and ensemble imputation respectively. As this study primarily investigates datasets with smaller sample sizes, we excluded deep learning-based MVI methods, as they tend to underperform in such cases and are black box algorithms [32]. Nevertheless, MICE and RF are strong performing alternatives, capable of handling complex missingness and small sample size settings [33]. MVI method descriptions are found in Supplementary Information. MVI was performed with all batches combined.

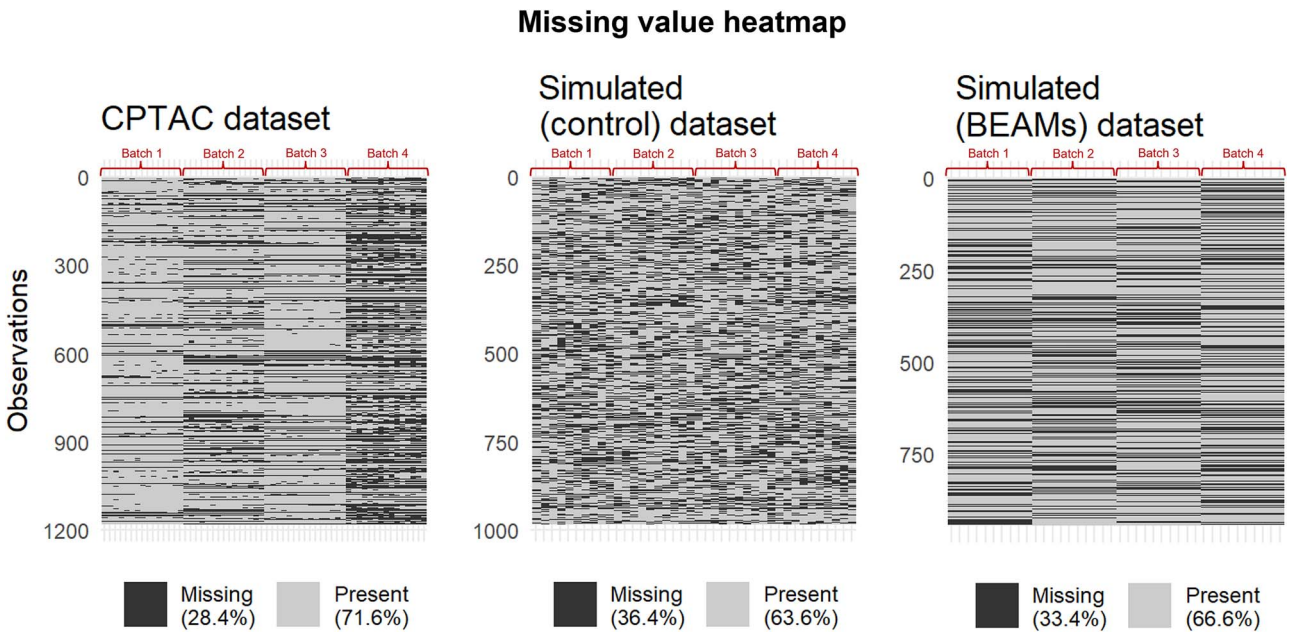

Figure 2. MV heatmaps of CPTAC and simulated (control and BEAMs) datasets. Black sections represent MVs while gray sections represent observed values. Real-world data (CPTAC dataset) appears to have a mixture of missingness from both control and BEAMs datasets, where MVs occur according to samples and batches respectively.

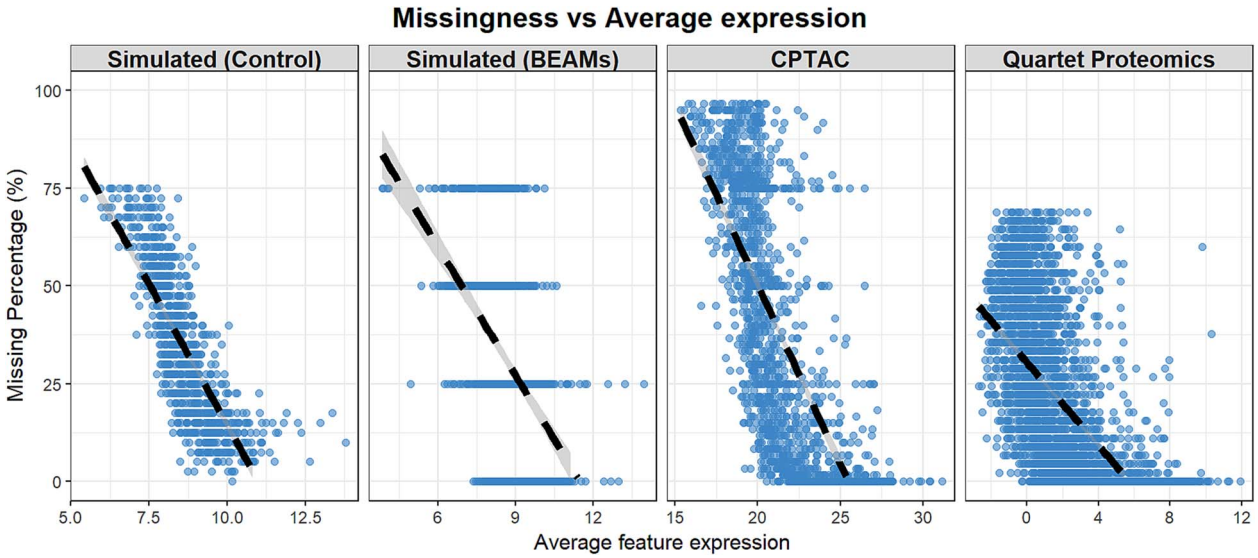

Figure 3. MV proportions against average feature expressions plotted for the simulated (control and BEAMs) datasets and real-world (CPTAC and quartet proteomics) datasets. The trend line indicates that all datasets contain MVs that are generally MNAR in nature.

Batch effect correction

ComBat—ComBat is a commonly used BECA due to its robustness to outliers and small sample sizes [3]. ComBat works on an empirical Bayes framework to estimate and standardize the mean and variance of different batches. ComBat was applied using the 'sva' R package, version 3.50.0 [5].

limma—limma fits a linear model to remove components associated with the user-supplied batch factor. We applied limma using the 'limma' R package, version 3.58.1 [4].

Statistical data analysis and visualization

Principal component analysis

PCA is a dimensionality reduction technique that identifies directions of greatest variance in the data, producing a set of linearly uncorrelated principal components (PCs) [10]. PCA is useful to evaluate how imputing BEAMs affect the preservation of data structures and BEC effectiveness. As strong class effects obscure batch clusters in the first few PCs, we removed true differentially expressed (DE) features prior to PCA.

For the CPTAC dataset, the ground truth was generated by performing PCA on the expression matrix with MV-laden proteins removed. This is necessary as PCA cannot be performed when MVs are present. We assume that with this filtering, the inter-sample relationships remain conserved. For simulations, the last iteration of simulated datasets was used.

Root mean squared error

Imputation accuracy was measured using the root mean squared error (RMSE), which represents the difference between each imputed value x_i and the true value y_i. Therefore, the lower the

RMSE, the better the imputation accuracy. RMSE can be calculated as follows:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{n}}$$

To assess RMSE in real-world datasets, we removed MV-laden features to obtain a complete dataset before introducing artificial MVs to be imputed. This allows us to know the ground truth values for RMSE calculation.

Hierarchical clustering and Pearson correlation

Hierarchical clustering is an unsupervised learning method that clusters data points based on dissimilarities in the data. Samples were clustered using the Euclidean distance dissimilarity measure. To enhance analysis, hierarchical clustering was used in combination with an inter-sample Pearson correlation coefficient heatmap to monitor detailed changes in the data structure. For the simulated datasets, the last iteration was used to perform this analysis.

Differential expression analysis

DEA was performed using unpaired *t*-tests for pairwise class comparisons, with *P*-values adjusted via the Benjamini–Hochberg procedure. Features with adjusted *P*-values <0.05 and absolute Log2 Fold-Change >0.5 were considered DE. With the true differential features known, we identified the true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). These allow us to calculate the True Positive Rate (also known as recall) and false discovery rate (FDR) of each dataset. Low FDR and high recall indicate better performances. These measures were not calculated for the real-world datasets due to the lack of a suitable BEAMs-negative control or ground truth. The formulas for these measures are shown below:

$$Recall = \frac{TP}{TP + FN}$$

$$FDR = \frac{FP}{FP + TP}$$

Results

Batch effect correction can go awry when batch effect associated missing values are imputed

Comparing the PCA visualizations of the imputed and corrected datasets against the ground truth, we can identify changes to the data structure resultant from imputing BEAMs. When ComBat was performed after MVI on the CPTAC dataset, we observed two distinct differences from the MV removed dataset (Fig. 4a). Firstly, in the KNN, SVD, MICE, and RF imputed datasets, samples clustered by class despite the absence of true DE proteins, suggesting that class differences had been incorrectly introduced. Secondly, Mean imputation hindered the BEC efficiency of ComBat.

We repeated the experiment on both simulated (control and BEAMs) datasets, finding similar outcomes to that observed with the CPTAC dataset. Here, the ground truth refers to the complete simulated dataset without MV insertion. Imputation on the simulated (control) dataset using Mean and MinProb showed relatively similar outcomes to the ground truth (Fig. 4b). Meanwhile, KNN, SVD, MICE, and RF imputations resulted in minor class clusters after ComBat correction (Fig. 4b). However, the same methods (except MICE) applied on the simulated (BEAMs) dataset resulted in more prominent class clusters after ComBat correction (Fig. 4c). These findings support our initial observations on the CPTAC dataset, where datasets imputed with KNN, SVD, and RF revealed class-associated clusters after BEC, despite the absence of true DE features in the data. In addition, Mean imputation on BEAMs similarly caused ComBat to struggle with BEC (Figs. 4a and c).

Increasing severity of batch effect associated missing values incurs greater errors

To further investigate the impact of BEAMs severity on MVI performance, we sectioned the simulated datasets by features into three mutually exclusive subgroups: non-BEAMs (features observed in all batches), moderate BEAMs (features missing in at least one batch), and severe BEAMs (features observed in only one batch). The accuracy of MVI methods was then assessed on both the moderate and severe BEAMs subgroups. Figure 5 shows that while all MVI methods appeared to react differently to moderate BEAMs, with some even producing lowered RMSE values, their performances collectively worsened with severe BEAMs. This was true for both CPTAC and simulated datasets. This indicates that varying BEAMs severities can indeed dictate the performance of MVI, regardless of the approach selected. We observed similar outcomes in the Quartet proteomics and metabolomics datasets (Fig. S1). We additionally tested limma's removeBatchEffect() function [4] to ensure that our findings were not specific to ComBat and found comparable results (Fig. S2).

Imputing batch effect associated missing values disrupts data integrity by influencing inter-sample correlations

Given the distorted PCA visualizations and poor imputation accuracy with BEAMs, we suspected that imputing BEAMs may affect the integrity of inter-sample correlations and sample clusters. To investigate this, we sectioned the CPTAC dataset and simulated (BEAMs) dataset according to the same subgroups as in the RMSE analysis. True DE features were then removed to focus on false effects. We under-sampled groups with more features to ensure that all three subgroups contained the same number of features. In this section, we focus on the less understood effects of BEAMs on MinProb imputation and SVD imputation for brevity (Fig. 6). A comparison of the effects of BEAMs on the remaining MVI methods is provided in the supplementary materials (Figs S4–S7).

Here, we use the subgroup of non-BEAMs features to approximate the ground truth as it requires little to no imputation. MinProb imputation on increasing BEAMs severities appeared to have minimal influence on how samples clustered, but inter-sample correlations increasingly deviated from the ground truth (Fig. 6, top panels). In addition, moderate BEAMs features imputed using SVD caused samples to cluster by class, especially after ComBat correction (Fig. 6, bottom panel). Like MinProb, severe BEAMs imputed with SVD resulted in stochastic inter-sample correlations both before and after ComBat correction, where samples seemed to form clusters that were unrelated to both batch and class factors. These findings were consistent in both CPTAC and simulated (BEAMs) datasets. For all MVI methods, data integrity seemed to worsen with increasing BEAMs severity (Figs S4–S7).

Imputing batch effect associated missing values can create the illusion of good performance

While our prior analyses show that imputing BEAMs can distort data structure, its impact on downstream outcomes remains unclear. Therefore, to determine if imputing BEAMs affects downstream analysis, we performed DEA on both simulated (con-

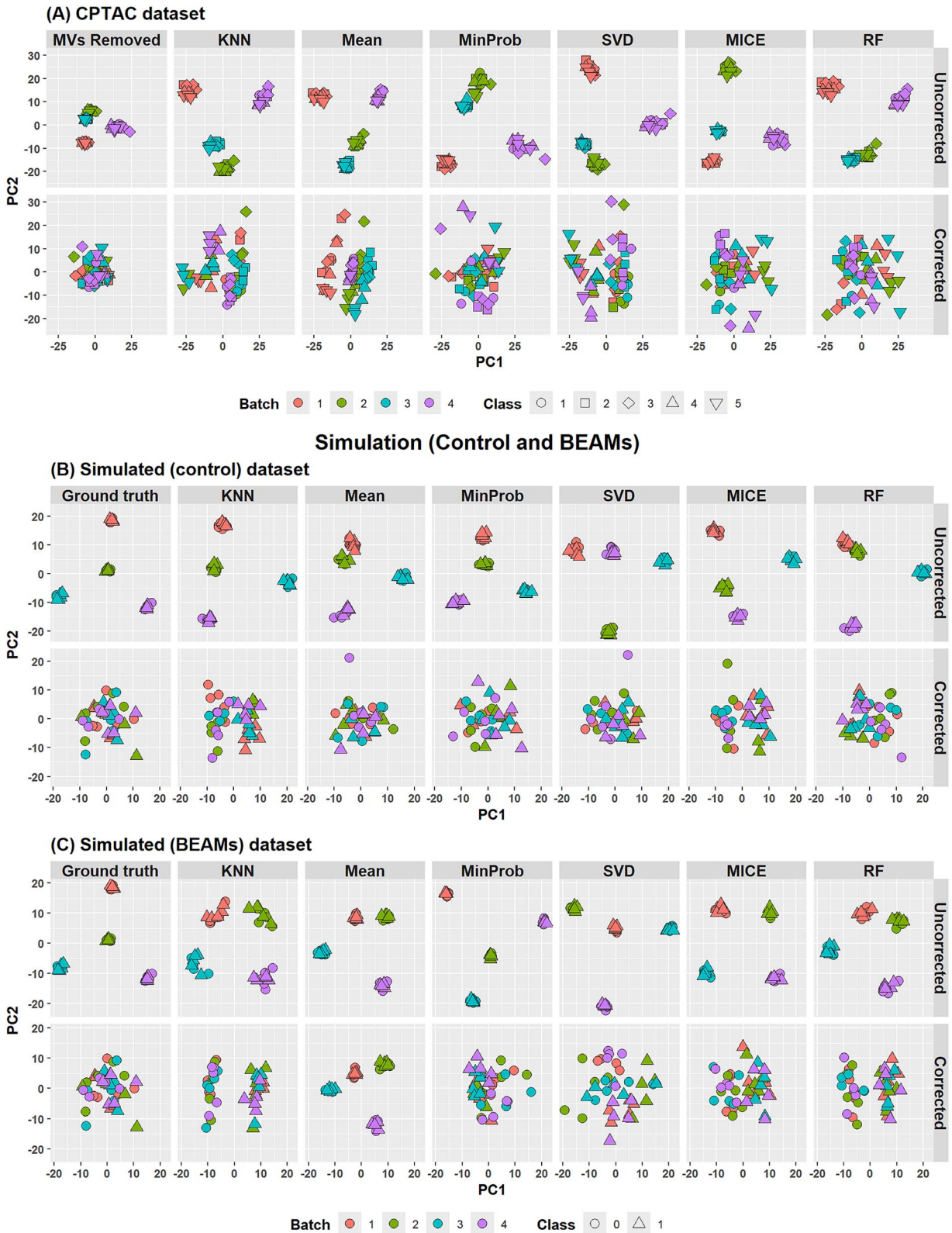

Figure 4. PCA visualization of the first and second PCs obtained from the (a) CPTAC datasets, uncorrected and corrected. In the 'MV removed' panels, features with MVs were not imputed, but instead removed prior to PCA. The final iteration of the simulated datasets was used to obtain the PCA visualizations of the (b) simulated (control) dataset, and the (c) simulated (BEAMs) dataset. Class-associated clusters were revealed after BEC, indicating that BEAMs may lead to class artifacts when imputed.

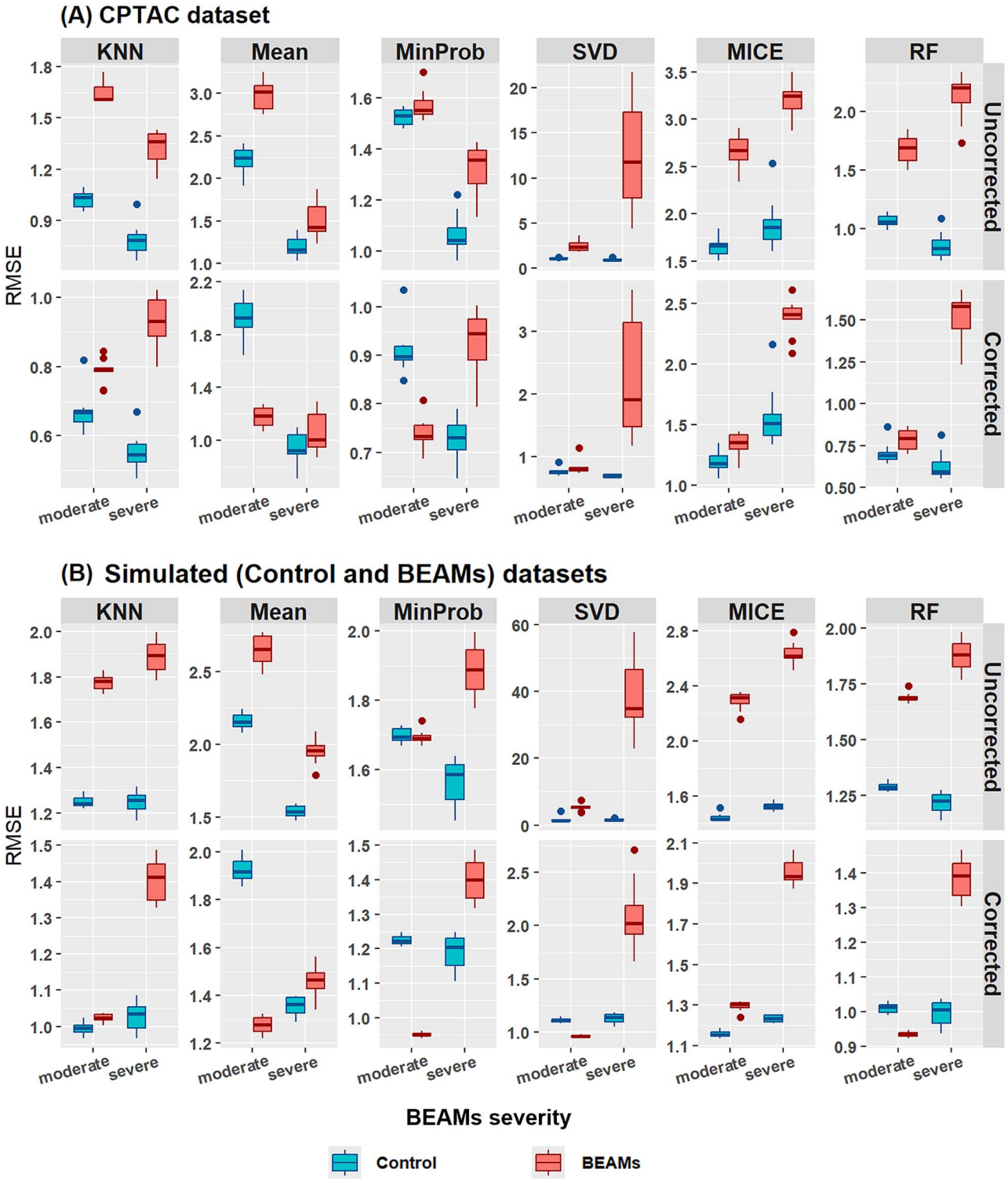

Figure 5. RMSE comparison between control and BEAMs missingness sectioned by features with varying BEAMs severity for both (a) CPTAC and (b) simulated datasets, across 10 iterations. Imputing severe BEAMs resulted in higher RMSE across all MVI methods.

trol and BEAMs) datasets using a student's *t*-test. The obtained *P*-values were adjusted through the Benjamini-Hochberg procedure. Figure 7 shows that the presence of BEAMs generally resulted in an increase in FDR during DEA, as opposed to when MVI was performed when non-BEAMs were present. This was particularly evidenced in datasets imputed by KNN, SVD, and RF. However, this was accompanied by an increase in recall across most MVI methods.

We then examined the adjusted *P*-value distributions of non-DE features to explain the rise in FDR (Fig. 8). Most imputed simulated (control) datasets showed left-skewed adjusted *P*-value distributions. In contrast, a large proportion of non-DE features

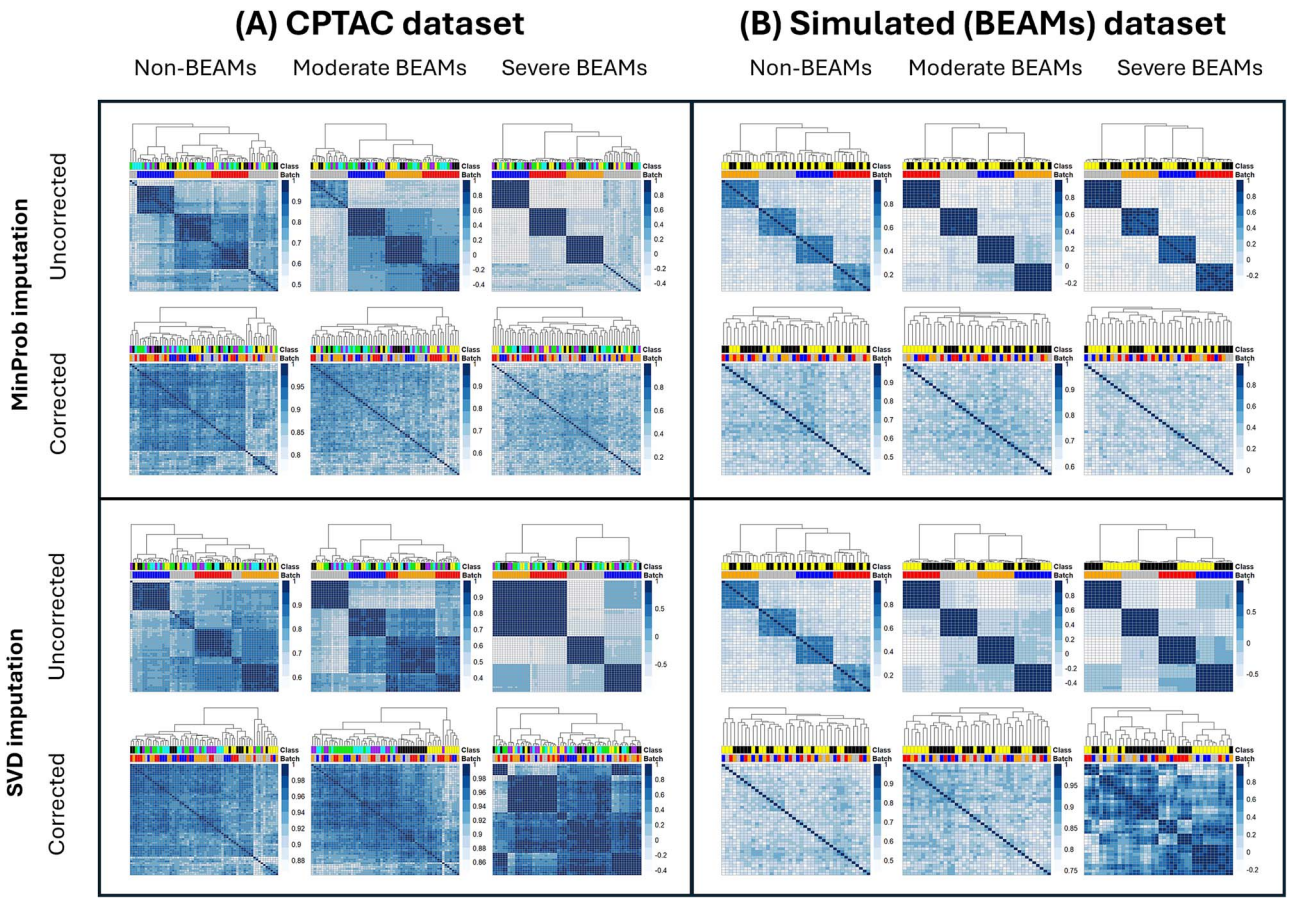

Figure 6. Inter-sample Pearson correlation coefficient heatmap and hierarchical clustering of (top) MinProb-imputed and (bottom) SVD-imputed (a) CPTAC dataset and (b) simulated (BEAMs) dataset. The data matrices were split into three subgroups of features according to the severity of BEAMs. The last iteration of simulated (BEAMs) dataset was used for this analysis. Inter-sample correlations increasingly deviated from the expected range when imputing BEAMs of greater severity.

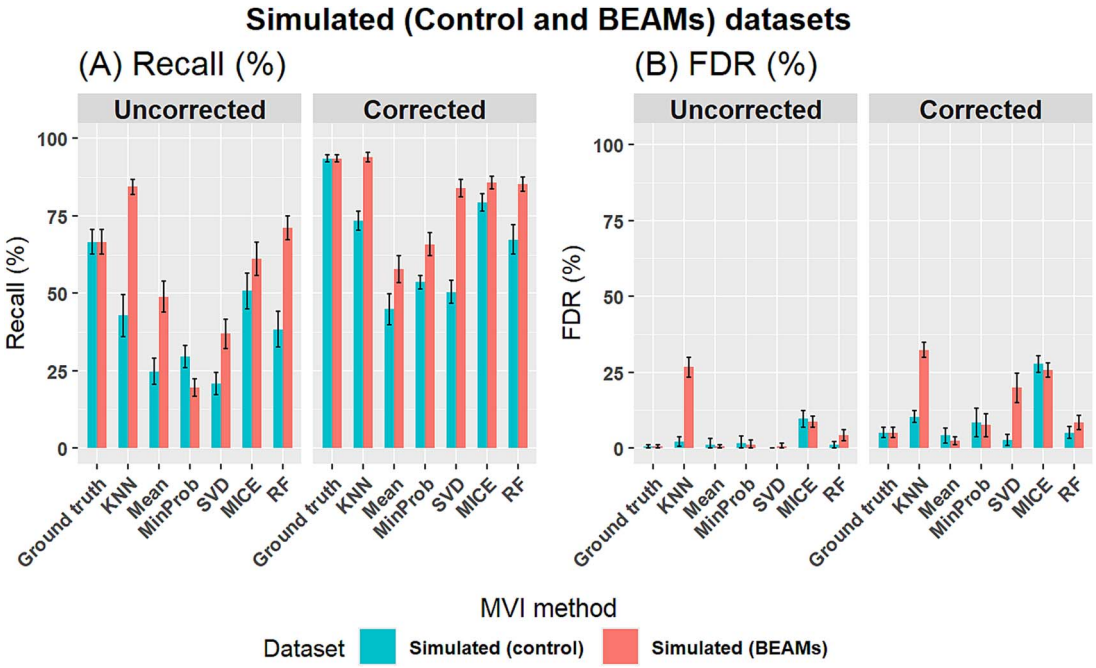

Figure 7. Simulated (control and BEAMs) datasets DEA results based on Student's pairwise *t*-test: (a) recall and (b) FDR. Results were obtained across 10 iterations of simulations. Error bars represent mean +/− sd. Imputing BEAMs led to increased FDR, particularly in KNN, SVD, and RF. However, this coincided with higher recall scores.

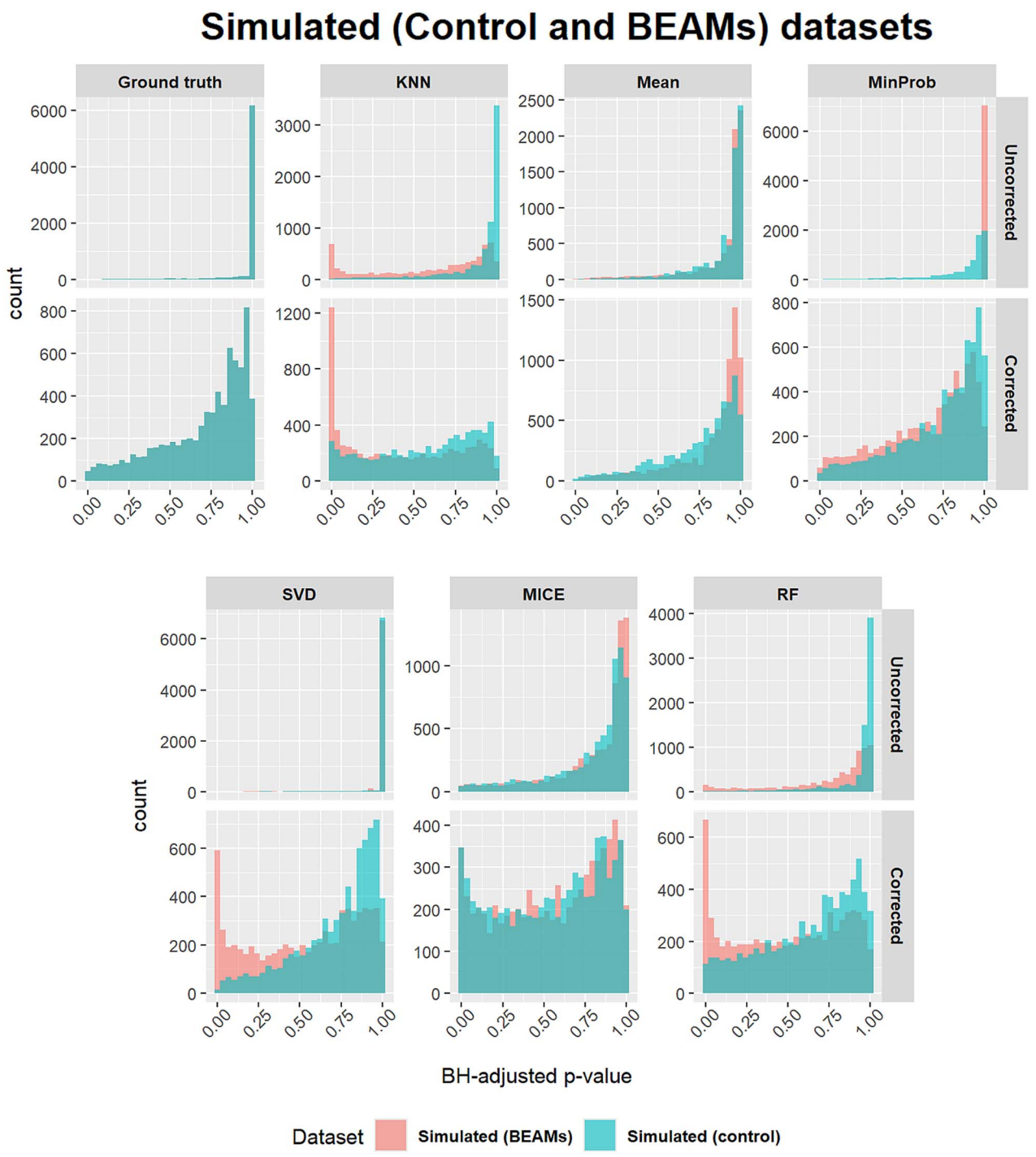

Figure 8. Distribution of Benjamini–Hochberg adjusted *P*-values obtained from Student's *t*-test >10 iterations of simulated (control and BEAMs) datasets, both uncorrected and corrected. Imputing BEAMs appeared to result in more statistically significant *P*-values.

were statistically significant in the simulated (BEAMs) datasets imputed using KNN, SVD, and RF. This implies that false class effects were introduced when BEAMs were imputed using these methods, which corroborates with our previous observations in Figs 4b and c.

To understand the spike in significant *P*-values in simulated (BEAMs) dataset imputed using KNN and RF, we randomly selected a DE feature from the KNN-imputed severe BEAMs subgroup and examined its expression values. Before simulating BEAMs, no class effects were visible (Fig. 9a). Upon introducing BEAMs, the only batch containing the feature appeared to exhibit a slight class difference that likely arose from noise (Fig. 9b). As KNN imputation identifies similar samples for estimation, a lack of same-batch observations causes all nearest neighbors to belong to different batches. This propagates the random variation from the observed batch to all other batches, creating a systematic pattern that is statistically significant (Fig. 9c). This was exacerbated when BEs were corrected, revealing explicit class disparities (Fig. 9d). A similar outcome was observed in the RF-imputed dataset (Fig. S8). While this phenomenon may not apply

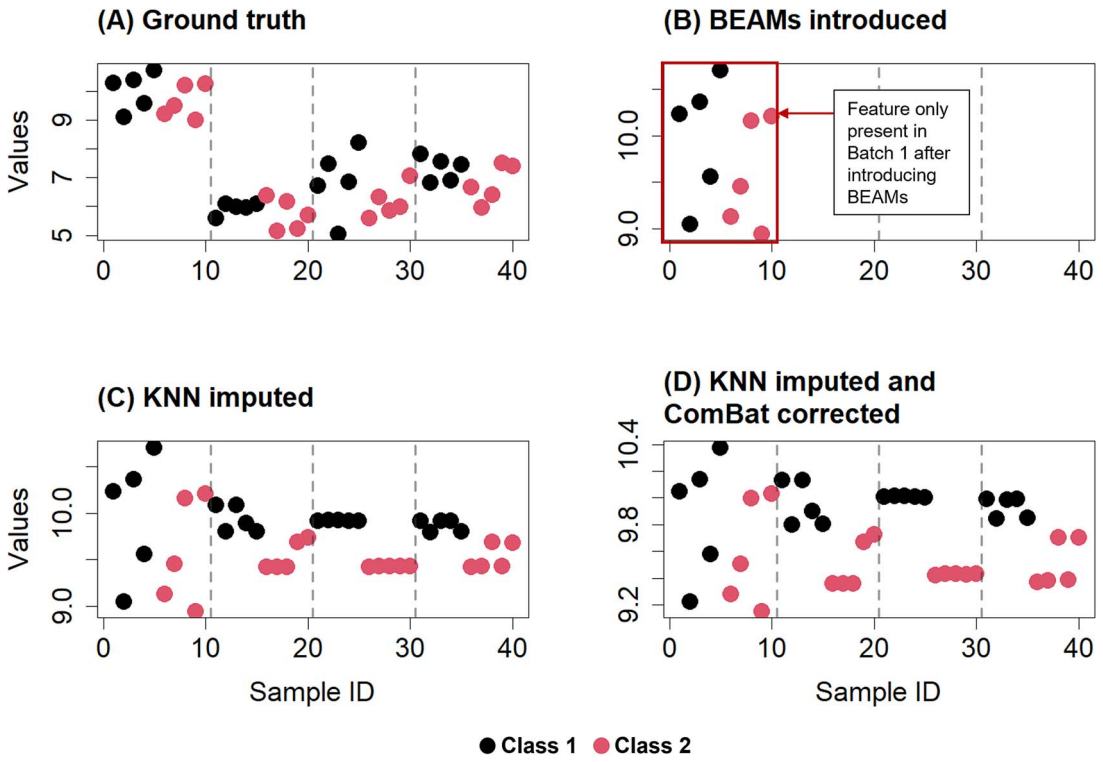

Figure 9. (a) A randomly sampled feature in the simulated (BEAMs) dataset before MV insertion. Dashed gray vertical lines segregate the sample IDs into four batches. (b) BEAMs were inserted into the feature, causing it to only be observed in batch 1. (c) Feature values after KNN imputation. (d) Feature values after KNN imputation and ComBat correction.

to other MVI methods operating on principles different from KNN and RF, it demonstrates the unreliability of compounding signals from small sample sizes through MVI.

Discussion

Data obtained from different batches often have varying coverage, which is the extent to which the genome or proteome are successfully read. Integrating these batches could result in BEAMs which, when imputed carelessly, can lead to a severe form of MVI-BEC confounding. Our findings demonstrate that mishandling BEAMs in data can mislead analysis, as downstream analysis may seemingly perform well despite haphazard alterations to the data structure post-imputation. All MVI methods assessed in this study led to spurious changes to either the visualized clusters, inter-sample correlations, or both, and performed poorly when met with severe BEAMs.

Our findings indicate that none of the MVI methods evaluated are suitable for handling BEAMs effectively. KNN and RF imputation, while improving statistical power by propagating signals from one batch to others, can lead to false results due to their sensitivity to data groupings. Imputation using Mean and MinProb, though less prone to errors, obscures differential expression patterns and impedes the detection of true positives, making them less desirable regardless of the MV distribution. Imputation using SVD has shown mixed findings in past studies [21, 34]. Our findings reassert its instability, showing that BEAMs are detrimental to SVD, producing grossly inflated RMSE values and spiked significant *P*-values. This is because SVD fundamentally alters the data structure by imputing based on matrix factorization using placeholder values, which can affect the overall reliability of the analysis.

Notably, increased severity of BEAMs led to decreased imputation accuracy across all MVI methods (Fig. 5). Furthermore, imputation of BEAMs of increasing severity using KNN, Mean, MinProb, and SVD caused inter-sample correlation coefficients to increasingly deviate from the expected range. This may indirectly affect inter-feature correlations, as sample correlations are driven by underlying feature expression patterns across samples. This can therefore influence feature selection, as demonstrated by the change in *P*-value distributions. This was also consistent with the observation of class artifacts being predominantly present in the severe BEAMs subgroup.

As discussed earlier, imputing BEAMs with KNN, SVD, or RF may erroneously create statistically significant differences in non-differential features by amplifying random observations into non-random effects. Even with large batch sample sizes, BEAMs-associated features can mimic small-sample behaviors, introducing small-sample biases [35] and potential false positives (Fig. 8d). Instead of batch sample sizes, more consideration should be placed on the number of batches present in the dataset. This is because the number of batches can dictate the occurrence of BEAMs. For example, let p be the probability of detecting a feature in a batch in a dataset with N batches. We assume that p is independent of batch or the theoretical expression value. As a moderate BEAMs feature is defined as a feature which is missing in at least one batch, the probability of having a moderate BEAMs feature is:

$$P\left(\textit{Moderate BEAMs}\right) = 1 - P\left(\textit{Detected in all batches}\right) = 1 - p^N$$

In the same vein, the probability of severe BEAMs, which are defined as features observed in only one batch, can be

calculated as:

$$P(\textit{Severe BEAMs}) = N \times p \times (1-p)^{N-1}$$

As N increases, the probability of moderate BEAMs increases while the probability of severe BEAMs decreases. Therefore, under these naïve assumptions, we can assume that increasing number of batches reduces the presence of severe BEAMs at the expense of incurring moderate BEAMs.

This study underscores the necessity of rigorous evaluation of both MVI and BEC processes when BEAMs are present in the data. Relying on a single metric (e.g. recall) to assess the performance of MVI and BEC methods can create a false sense of confidence, potentially leading to erroneous biological interpretations. While several metrics should be used for evaluation, it may also be useful to section the data feature-wise according to BEAMs severity to better understand the impact of BEAMs in our data. In general, our findings indicate that severe BEAMs are poorly imputed using most MVI methods. The silver lining to this is that moderate BEAMs are less detrimental, and that the influence of BEAMs on MVI can be somewhat controlled by removing severe BEAMs features prior, ensuring that every feature is represented by multiple batches.

We evaluated six MVI methods to understand the effect of BEAMs on MVI. However, many other MVI methods remain unexplored, particularly, platform-specific MVI methods. Given the popularity of the methods we evaluated, BEAMs-related issues are likely to have affected data analysis in the research community. Our findings highlight non-negligible interactions between MVI and BEC in the presence of BEAMs. This is also one of the first instances in the literature to provide an initial understanding of how BEAMs may affect MVI. Our findings may guide the development of new methods toward better BEAMs management.

Simulations were used to allow direct comparisons between BEAMs and non-BEAMs conditions, improving benchmarking explainability. However, a drawback is that it lacks true biological complexity. To our knowledge, this study is the first to model BEAMs. Since the occurrence of BEAMs is poorly understood, we modified a known MV simulation method [23] to simulate MVs. Based on our observations in real-world datasets, we assumed that BEAMs are predominantly MNAR in nature (Fig. 3). Nevertheless, we observed similar results when MCAR was the dominant mechanism (Fig. S3). Further refining simulations to better reflect real data could enhance analysis but also risks overfitting to the specific datasets.

In future work, we may induce BEAMs into a wider variety of data types and data sources. This can provide additional depth and breadth for parameters that can be incorporated into more sophisticated simulation work, that in turn, produces more realistic ground truth data useful for the development of approaches for tackling BEAMs. Additionally, due to the known risks of diluting class effect signals with global normalization, as discussed in Zhao *et al.* [25], we did not test alternative normalization methods in this study. However, we acknowledge this is an interesting area for future investigation.

In summary, BEAMs are incurred when batches with varying degrees of coverage are integrated. BEAMs may worsen imputation accuracy, potentially leading to inflated statistical significance and disrupted inter-sample correlations. Our findings underline the importance of proper handling of BEAMs in high-throughput biomedical data. A temporary solution to handle BEAMs is to simply remove the features from subsequent analysis. However, we emphasize the need for more sophisticated strategies that do not rely on cross-sample inferences, as these would be extremely beneficial for increasing the number of features available for analysis, especially on platforms with naturally high degrees of missingness.

Key Points

- Data integration can incur batch effect associated missing values (BEAMs) due to different levels of feature coverage.
- Missing value imputation methods perform poorly when BEAMs are present in the data.
- Development of novel strategies to specifically handle BEAMs are necessary to improve the analysis of integrated high-throughput biomedical data

Author contributions

Harvard Wai Hann Hui (Performed analyses, Developed figures, Wrote the manuscript), Wei Xin Chan (Conceived the data simulation codes, Contributed to research design, Wrote the manuscript), and Wilson Wen Bin Goh (Conceptualized, Supervised, Provided critical feedback, Wrote the manuscript).

Supplementary data

Supplementary data is available at Briefings in Bioinformatics online.

Conflict of interest: None declared.

Funding

This research/project is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 (RS08/21 and RT11/21).

Data availability

The scripts used to generate the results in this study can be accessed at https://github.com/HarvardHui/BEAMs.

All codes were written in the R programming language (version 4.3.2) within the RStudio integrated development environment (version RStudio 2023.12.0 + 369), adhering to relevant guidelines and regulations.

The CPTAC study 6 data can be obtained from: https://pdc.cancer.gov/pdc/TechnologyAdvancementStudies, under the identifier: PDC000006.

The Quartet proteomics and metabolomics datasets can be obtained from the paper by Yu *et al.* [29].

References

1. Leek JT, Scharpf RB, Bravo HC. *et al.* Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet* 2010;**11**:733–9. https://doi.org/10.1038/nrg2825.
2. Goh WWB, Wang W, Wong L. Why batch effects matter in omics data, and how to avoid them. *Trends Biotechnol* 2017;**35**:498–507. https://doi.org/10.1016/j.tibtech.2017.02.012.

3. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**:118–27.
4. Ritchie ME, Phipson B, Wu D. *et al.* Limma powers differential expression analyses for RNA-sequencing and microarray studies. *Nucleic Acids Res* 2015;**43**:e47. https://doi.org/10.1093/nar/gkv007.
5. Leek JT, Johnson WE, Parker HS. *et al.* The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics* 2012;**28**:882–3. https://doi.org/10.1093/bioinformatics/bts034.
6. Gagnon-Bartsch JA, Speed TP. Using control genes to correct for unwanted variation in microarray data. *Biostatistics* 2012;**13**:539–52. https://doi.org/10.1093/biostatistics/kxr034.
7. Oytam Y, Sobhanmanesh F, Duesing K. *et al.* Risk-conscious correction of batch effects: maximising information extraction from high-throughput genomic datasets. *BMC Bioinformatics* 2016;**17**:332.
8. Kong W, Hui HWH, Peng H. *et al.* Dealing with missing values in proteomics data. *Proteomics* 2022;**22**:2200092.
9. Goeminne LJE, Sticker A, Martens L. *et al.* MSqRob takes the missing hurdle: uniting intensity- and count-based proteomics. *Anal Chem* 2020;**92**:6278–87. https://doi.org/10.1021/acs.analchem.9b04375.
10. Jolliffe IT. *Principal Component Analysis for Special Types of Data*. New York: Springer, 2002;338–72.
11. Gardner ML, Freitas MA. Multiple imputation approaches applied to the missing value problem in bottom-up proteomics. *Int J Mol Sci* 2021;**22**:9650.
12. Webb-Robertson B-JM, Wiberg HK, Matzke MM. *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. https://doi.org/10.1021/pr501138h.
13. Hui HWH, Kong W, Peng H. *et al.* The importance of batch sensitization in missing value imputation. *Sci Rep* 2023;**13**:3003.
14. Čuklina J, Lee CH, Williams EG. *et al.* Diagnostics and correction of batch effects in large-scale proteomic studies: a tutorial. *Mol Syst Biol* 2021;**17**:e10240.
15. Brenes A, Hukelmann J, Bensaddek D. *et al.* Multibatch TMT reveals false positives, batch effects and missing values. *Mol Cell Proteomics* 2019;**18**:1967–80. https://doi.org/10.1074/mcp.RA119.001472.
16. Matafora V, Corno A, Ciliberto A. *et al.* Missing value monitoring enhances the robustness in proteomics quantitation. *J Proteome Res* 2017;**16**:1719–27.
17. Paulovich AG, Billheimer D, Ham A-JL. *et al.* Interlaboratory study characterizing a yeast performance standard for benchmarking LC-MS platform performance. *Mol Cell Proteomics* 2010;**9**:242–54.
18. Li WV, Li JJ. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nat Commun* 2018;**9**:997.
19. Hicks SC, Townes FW, Teng M. *et al.* Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics* 2018;**19**:562–78. https://doi.org/10.1093/biostatistics/kxx053.
20. Välikangas T, Suomi T, Elo LL. A comprehensive evaluation of popular proteomics software workflows for label-free proteome quantification and imputation. *Brief Bioinform* 2017;**19**:1344–55.
21. Wei R, Wang J, Su M. *et al.* Missing value imputation approach for mass spectrometry-based metabolomics data. *Sci Rep* 2018;**8**:663.
22. Liu M, Dongre A. Proper imputation of missing values in proteomics datasets for differential expression analysis. *Brief Bioinform* 2021;**22**:bbaa112.
23. Jin L, Bi Y, Hu C. *et al.* A comparative study of evaluating missing value imputation methods in label-free proteomics. *Sci Rep* 2021;**11**:1–11.
24. Xu X, Yu X, Hu G. *et al.* Propensity score matching enables batch-effect-corrected imputation in single-cell RNA-seq analysis. *Brief Bioinform* 2022;**23**:bbac275.
25. Zhao Y, Wong L, Goh WWB. How to do quantile normalization correctly for gene expression data analyses. *Sci Rep* 2020;**10**:15534.
26. Zheng Y, Liu Y, Yang J. *et al.* Multi-omics data integration using ratio-based quantitative profiling with quartet reference materials. *Nat Biotechnol* 2024;**42**:1133–49. https://doi.org/10.1038/s41587-023-01934-1.
27. Tian S, Zhan D, Yu Y. *et al.* Quartet protein reference materials and datasets for multi-platform assessment of label-free proteomics. *Genome Biol* 2023;**24**:202.
28. Zhang N, Zhang P, Chen Q. *et al.* Quartet metabolite reference materials for assessing inter-laboratory reliability and data integration of metabolomic profiling. bioRxiv 2022. https://doi.org/10.1101/2022.11.01.514762.
29. Yu Y, Zhang N, Mai Y. *et al.* Correcting batch effects in large-scale multiomics studies using a reference-material-based ratio method. *Genome Biol* 2023;**24**:201.
30. Lazar C, Gatto L, Ferro M. *et al.* Accounting for the multiple natures of missing values in label-free quantitative proteomics data sets to compare imputation strategies. *J Proteome Res* 2016;**15**:1116–25. https://doi.org/10.1021/acs.jproteome.5b00981.
31. Karpievitch Y, Stanley J, Taverner T. *et al.* A statistical framework for protein quantitation in bottom-up MS-based proteomics. *Bioinformatics* 2009;**25**:2028–34. https://doi.org/10.1093/bioinformatics/btp362.
32. Sun Y, Li J, Xu Y. *et al.* Deep learning versus conventional methods for missing data imputation: a review and comparative study. *Expert Systems with Applications* 2023;**227**:120201. https://doi.org/10.1016/j.eswa.2023.120201.
33. Bramer LM, Irvahn J, Piehowski PD. *et al.* A review of imputation strategies for isobaric Labeling-based shotgun proteomics. *J Proteome Res* 2021;**20**:1–13. https://doi.org/10.1021/acs.jproteome.0c00123.
34. Kokla M, Virtanen J, Kolehmainen M. *et al.* Random forest-based imputation outperforms other methods for imputing LC-MS metabolomics data: a comparative study. *BMC Bioinformatics* 2019;**20**:492.
35. Wang W, Sue AC-H, Goh WWB. Feature selection in clinical proteomics: with great power comes great reproducibility. *Drug Discov Today* 2017;**22**:912–8. https://doi.org/10.1016/j.drudis.2016.12.006.

Briefings in Bioinformatics, 2025, **26(2)**, bbaf168
https://doi.org/10.1093/bib/bbaf168
Case Study