# Why Batch Sensitization is Important for Missing Value Imputation

**Priscila Yun Qian Sun**
Nanyang Technological University
**Wilson Wen Bin Goh** ( wilsongoh@ntu.edu.sg )
Nanyang Technological University

**Research Article**

**Keywords:** Batch effect, Missing value imputation, Bioinformatics, Statistics

**Posted Date:** February 25th, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-1328989/v1



**Additional Declarations:** No competing interests reported.

**Version of Record:** A version of this preprint was published at Scientific Reports on February 21st, 2023.
See the published version at https://doi.org/10.1038/s41598-023-30084-2.

# Abstract

Data analysis is complex due to a myriad of technical problems. Amongst these, missing values and batch effects are particularly endemic. Although many methods have been developed for missing value imputation (MVI) and batch correction respectively, no study has directly considered the confounding impact of MVI on downstream batch correction. This is surprising as missing values are imputed during early pre-processing while batch effects are mitigated during late pre-processing, prior to functional analysis. The problem is that unless actively managed, MVI approaches generally ignore the batch covariate, with unknown consequences. We examine this problem by modelling 3 imputation strategies: global (M1), self-batch (M2) and cross-batch (M3) using simple matrix simulations, proteomics and genomics data. Considering batch covariates (M2) is important, resulting in enhanced batch correction and lower statistical errors. However, M1 and M3 are insidious: global and cross-batch averaging results in batch-effect dilution, with concomitant and irreversible increase in intra-sample noise. This noise is unremovable via batch correction algorithms and produces false positives and negatives. Hence, careless imputation in the presence of non-negligible covariates such as batch effects is costly.


# Introduction

Missing values (MV) and batch effects (BE) are both endemic problems in high-dimensional biological data analyses. The former relates to particular information points being present in some samples but not others [1] while the latter relates to technical sources of biases that may confound the true signal-of-interest [2]. To date, a large number of missing value imputation (MVI) and batch-effect correction algorithms (BECAs) have been developed. Although neither are considered fully “solved” problems, many would concur that they are at least “manageable”, provided appropriate conditions are fulfilled. However, MV and BE are not mutually exclusive problems. Although typically treated separately, there are in fact, temporal dependencies between the two: In general, MVs are imputed during early phases of pre-processing (producing a pseudo-complete data matrix) while batch effects are dealt with during late pre-processing (on the pseudo-complete matrix), prior to functional analysis. This temporal dependency means that the efficacy of the batch correction is dependent on how MVI was performed. Suppose we know early on that a batch (or some other important and non-negligible) co-variate exists, we hypothesize it would make more sense to impute using this information early on.

Despite the apparentness of this hypothesis, a search in current literature using “Missing Value Imputation” and “Batch Effect” did not reveal any studies (or mention) examining confounding effects between the two. A few review articles did emerge, but discussed the topics as entirely separate entities (as a checklist of considerations in data processing).

Hence, this work is the first to explore and evaluate how batch sensitization in MVI impacts batch correction (which subsequently impairs the ability to identify correct gene targets). Where both batch effects and missing values are present in data, we recommend caution, and to ensure that the batch factor is taken into consideration early on in the data processing stage.

# Batch-sensitized missing value imputation improves batch correction

RMSE is a measure of the difference between imputed data and true data. The lower the RMSE the better the imputation accuracy. We first evaluate RMSE across 4 BECAs (ComBat, BMC, Harman and SVA) on mixed batch effects (Fig. 2.1).

RMSE results show that M2 produces the lowest RMSE across BECAs (except for SVA where batch correction is unexpectedly unstable) (Fig. 2.1). This suggests M2 is a superior strategy. However, it is unknown if this extends toward real biological data where complex inter-correlations amongst variables exist. Hence, we consider proteomics and genomics data for our subsequent analyses.

In addition, we noted that in general, most BECAs (with the exception of SVA) respond favourably to M2. Amongst these, ComBat is the most widely used [3–5] and also a strong performer across many benchmark evaluations [6]. Our goal is not to identify the best BECA, but rather to demonstrate that batch co-variate sensitization is necessary for effective MVI. And so, for subsequent analyses, we only report results based on ComBat.

In agreement with simulations (Fig. 2.1), M2 provides the best imputation accuracy for both proteomics and genomics data due to having the lowest RMSE (Fig. 2.2 & 2.3).

# Batch Correction

gPCA delta is a relative estimation of batch proportion in the data. The lower the gPCA delta, the smaller the relative degree of batch-correlated separation in data, and hence, less batch effects (scaled between 0 to 1 regardless of actual batch effect magnitude). Interestingly, although RMSE suggests M2 is the best strategy that yields a more similar matrix to original, gPCA suggests that batch effects are more persistent given M2 (Fig. 3.1). This may be attributed to the high percentage of missing values in our simulations (i.e., 50%). Hence, to check whether gPCA determined batch effects are correlated with the number of missing values, we change the missingness from 10%, 20%, 30% and 40% (and keep the results for 0% and 50% from the initial simulation) for M2. Our hypothesis was proven correct: after batch correction, remnant batch in M2 is attenuated given lower percentages of missing values (Fig. 5). Subsequent analyses based on PCA and data distribution checks post imputations and batch correction revealed that gPCA was misleading –– the apparent undetectability of a batch effect was because of batch signal mixing due to M1 and M3 strategies, resulting in noise generation (see Genomics and Proteomics analysis).

Similar to pure simulations (Fig. 3.1), M2 does not perform well for proteomics based on batch correction given gPCA delta (Fig. 3.2). Unlike simulations and proteomics (Figs. 3.1 & 3.2), M2 performed on genomics data has somewhat good batch effect reduction (i.e., lowest gPCA delta), however, it is still quite comparable to M1 (Fig. 3.3). We found this highly suspicious as M1 and M3 are reporting much lower gPCA deltas than expected while also exhibiting high dissimilarities to the original matrix (high

RMSE). To unravel this, we checked the principal components analysis (PCA) scatterplots based on the first 2 principal components (PCs) (Fig. 6).

Given the first 2 PCs, batch effects across M1 to M3 appears well resolved for both proteomics (Fig. 6.1C, 6.1D & 6.1E) and genomics (Fig. 6.2C, 6.2D & 6.2E). Despite reporting higher gPCA levels for M2, samples appear well-mixed with no apparent batch effects for all imputation strategies (M1 to M3) given the first two principal components (PC1 and PC2). We do note that for M2, some same-batch samples do appear more clustered following batch correction (Fig. 6.1D).

# M1 and M3 inflates intra-sample variance

Batch-corrected M1 to M3 samples appear fairly well-mixed given PCA analysis. It is possible that gPCA may be misreporting the extent of residual batch effects. However, this does not explain the high RMSE observed in M1 and M3 or the low RMSE given M2. Hence, we devised a simple approach to check sample distributions in original data, and also given M1-M3 imputation strategies. We expect high variance in samples following M1 and M3 imputation, thereby giving rise to high RMSE. Our hypothesis was proven correct: while M2 preserved similar sample variances to original data, M1 and M3 variances were grossly inflated (Fig. 7). To us, this means that M1 and M3 imputation strategies have traded batch effects for noise. This explains the high RMSE and low gPCA observations.

# Power and Effect Size

Amongst the 3 imputation strategies, M2 provides the highest statistical power (i.e., the highest probability of detecting class effects when it does exist) after batch correction. However, we noted (and with good reason; see **Discussion**), it will never be as high as original batch corrected data with no missing values (with the exception of BMC) (Fig. 4). This suggests that imputation strategies are imperfect, and information loss occurs anyway.

Nonetheless, we do note that for genomics simulation in particular, statistical power for M2 after batch correction seems to be cross comparable to the original batch corrected data (Table 1). We hypothesize that this is because sample size of genomics data is large enough such that M2 is able to make meaningful imputations. This intuition turns out to be correct as after reducing the genomics data to a 20 x 20 matrix (similar to our pure simulation dimensions), M2 batch corrected no longer performs as well as batch corrected (Figure S11).

Table 1
Power for Genomics Simulation

| Variable | Power (2 d.p.) |
|---|---|
| base power | 1.00 |
| batch | 1.00 |
| batch corrected | 1.00 |
| m1 batch | 0.96 |
| m1 batch corrected | 0.99 |
| m2 batch | 0.84 |
| m2 batch corrected | 1.00 |
| m3 batch | 0.70 |
| m3 batch corrected | 0.70 |

Although M2 provides the best power, we hypothesize that imputation can dilute class effects as there are in fact, two classes within each batch. This intuition turns out to be correct as the t-statistics associated with M1, M2, M3 are generally lower than the original (without missing values) (Fig. 8 & S7).

We also noted (Fig. 4) that BMC and SVA do not work in these simulations where power is negatively affected even in original data. BMC and SVA do not work well on multiplicative-only and additive-only batch effect scenarios either (Figure S3). As neither methods are often reported as optimal in literature, nor are they able to satisfactory restore pre-batch performances in original data, we do not consider these for further analyses on real data.

One may expect therefore, that imputation based on the same class and batch may yield better results (we term this approach M2.1; see Figure S9). However, we find that this was not exclusively so. While M2.1 did indeed yield t-statistics more similar to original data, it did not outperform M2 in terms of RMSE, gPCA and power.

# Precision, Recall, False Positive Rate (FPR), False Discovery Rate (FDR)

Power deals with false negatives. We also need to consider mistakes stemming from false positives. We evaluate performances based on precision (proportion of correct features amongst selected features) and recall (proportion of correct features amongst all correct features), and also the false positive rate (FPR; the proportion of false positives stemming from wrong features) and false discovery rate (FDR; the proportion of wrong features amongst selected features). The results show that after batch correction, M2 performs the best with highest precision and recall and lowest FPR and FDR among post MVI data. What is especially notable is that M2 batch corrected seems to perform quite comparably to batch

correction (i.e. data without missing values) (Fig. 9). This is because sample size of genomics data is large enough for M2 to make meaningful imputations (same reasoning as to why, for genomics simulations, statistical power for M2 after batch correction is cross comparable to the original batch corrected data; see **Results: Power and Effect Size**).

## Discussions

# RMSE is more informative than gPCA delta

Although it does not estimate batch effects directly, RMSE is a more informative measure than gPCA delta. gPCA delta estimates batch effects only --- and so, fails when batch effects are diluted (Fig. 3). In our analyses, M1 and M3 approaches report lower batch effects than M2 but also has concomitant increase in intra-sample variances. If we were to believe gPCA alone, then we would have been misled into thinking M1 and M3 are reasonable imputation strategies.

In our simulations, we are able to measure imputation accuracy via RMSE, and so, we can compare how similar the imputed matrix is to the original. RMSE suggests that in spite of lower reported gPCA delta, M1 and M3 imputed matrices are very different from original data where no MVs existed. In contrast, M2 imputed matrices are comparatively similar to original data (Fig. 2).

The batch effect is diluted and cannot be detected via gPCA nor PCA scatterplots while intra-sample variance increased, resulting in high dissimilarity from original data. Hence, the RMSE is more informative than gPCA delta and the PCA scatterplots.

Unfortunately, on real data, there is no original data reference, and so RMSE cannot be deployed in practical scenarios. However, our first recommendation is that estimates of batch effects, whether summary (gPCA delta) or visual (PCA scatterplots), needs to be considered carefully: The lack of apparent batch effects in data, does not mean data quality is pristine.

# M1 and M3 trades batch effects for noise and this conversion is irreversible

While batch effects appear to be “mitigated” (Fig. 6), M1 and M3 result in increased noise (i.e., larger interquartile range) in the data (Fig. 7 & S10). Hence, it is sensible to impute missing values by M2 to avoid introducing additional noise into the data.

A very important observation is that in Fig. 7, intra-sample variance increases dramatically in M1 and M3 imputed samples (particularly so for proteomics data). This increased variance does not decrease post-batch correction, meaning it is no longer recognized as a batch effect, and so is not removed. In a practical setting, this also means that should you obtain the imputed matrix generated under wrong assumptions e.g. did not consider the batch co-variate, it is too late to apply a batch correction algorithm. This is an important example why deliberate and correct pre-processing strategies are very important.

**Drop in test statistic estimations is expected. But should we use both batch and class sensitization strategies in imputation?**

The value of the t-statistic and accompanying degrees-of-freedom (dependent on the actual number of non-missing values) determines the statistical p-value. In general, the t-statistics of M1, M2, M3 are lower than batch corrected (Fig. 8 & S7). The t-statistics indicate a drop in effect size estimations for all imputation strategies (M1, M2, M3). Therefore, even though M2 gives the highest power for post MVI data after batch correction (Fig. 4), the t-statistic distribution is appreciably different between imputed and original data.

The t-test can be divided into 2 components –– the numerator is expressed as the difference of means, which is a direct proxy for effect size. The denominator expresses uncertainty on this estimation of effect size and is related to the combined variances amongst samples. By dissecting both the numerator and denominator, it appears that the chief contributor towards reduction in the test-statistic, is due to increased variance due to imputation strategy (Figure S8)

We considered a more elaborate strategy incorporating both class and batch co-variates. While this strategy did improve the t-statistic distribution, it did not perform as well as M2 in terms of RMSE, gPCA and power.

# Imputation methods are no substitute for complete data

In spite of improved performances given more reasonable imputation assumptions, the performance of imputed matrices does not come close to original data without missing values (Fig. 4 & S11). However, we do recognize that we are considering a rather drastic scenario with 50% data loss. However, such high data losses are not unheard of in biological scenarios. In proteomics, 20–30% missing values is usual [7]. In genomics data, completion is normally higher, although missing values do occur, especially at low abundance levels (this is a form of MNAR). If missing values aggregate mostly at low abundance levels with high coefficient of variances, imputation may not provide satisfactory outcomes. Some normalization strategies e.g. gene fuzzy scoring (GFS) advocate ignoring noise from lower abundance levels due to higher noise [8].

# Limitations and future work

This work illustrates the importance of batch-sensitization in MVI data processing. We often forget use-case limitations in the design of data processing techniques. Another related example is the necessity for class-sensitization in widely used normalization methods such as quantile normalization [9].

Despite the simplicity of our simulations, the results are consistent. Future work may take into account other forms of MVs such as MAR and MNAR, which are known to be particular impacts on downstream analyses [10]. Amongst MVIs, we have opted for the simplest mean-based imputation method which is purely univariate. Global methods and other more sophisticated methods are not explored here. But we expect that these approaches should also be compatible with M2.

We also did not perform probabilistic or random value imputation. These methods also can add noise and unpredictability into our models. Presumably, it will take more simulations for the results to converge, without adding much value.

Our main aim is to make the simple point that considering the batch covariate (if known) early on is important. We expect this fundamental consideration will also hold true given any other MVI approach.

## Conclusions

Missing data is pervasive in data; and imputing without considering batch factors (batch sensitization) produces confounding effects. We show that M2 (batch-sensitized) gives the best imputation accuracy after batch correction. We also show that imputation by M1 (global) and M3 (cross-batch) introduce noise, which contribute directly towards false positives and false negatives. These results are consistent given various batch effect simulation strategies. We conclude that performing MVI without considering carefully important co-variates (such as batch effects) can mislead.

## Materials And Methods

All methods were performed in accordance with the relevant guidelines and regulations.

# Batch Effect Correction Algorithms (BECA)

# Combat

Combat is a widely used batch effect correction algorithm. It involves using an Empirical Bayes (EB) method to estimate the Location (mean) and Scale (variance) model parameters. These EB estimates are then used to adjust the data for batch effects. Combat is known to be robust to outliers and perform well for small sample sizes [11].

# Surrogate Variable Analysis (SVA)

SVA is a batch correction method that is based on matrix factorisation. It assumes batch effects are induced by unmodeled factors. SVA borrows information across samples to estimate the large-scale effects of all unmodeled factors from the data. Sources of variation induced by unmodeled factors can then be removed, thereby removing batch effects [12].

# Harman

Harman is a batch correction method that is based on Principal Component Analysis (PCA) and constrained optimisation. It removes batch effects from datasets, with the constraint that the probability of overcorrection (i.e., removing genuine biological signal along with batch noise) is kept to a certain limit. In our case, the confidence limit is set to 0.95, which means that the probability of removing biological signals along with batch noise is 0.05 [13].

# Batch Mean Centering (BMC)

BMC corrects batch effect by subtracting the batch mean from the data. This is equivalent to zero-centering each batch. Thus, after BMC adjustment, the mean of all samples in each batch is zero [14].

# Batch Effect Detection Methods

### PCA

PCA is often used as a dimensionality reduction technique for data compression and visualisation (on 2D/3D scatter plots). It reduces high dimensional data into lower numbers of linearly uncorrelated variables known as principal components (PCs). The first PC has the highest variance, and the second PC has the next highest variance. Combined with scatterplots, PCA is not effective in detecting batch effects if batch effects are not amongst the top sources of variation (PCs 1 to 3). PCA is commonly performed on data matrix, X, alone [15].

# Guided PCA (gPCA)

A more informative version of PCA for detecting batch effects is gPCA, which is guided by a batch indicator matrix to look for batch effects in the data. Typically, gPCA is performed on $Y^TX$, where Y is a batch indicator matrix and X is the data matrix. gPCA provided 2 metrics, a delta, which is the proportion of total variance of the data that is induced by batch effects, and its associated p-value. Both delta and its associated p-value range from 0 to 1. If delta is nearer to 1, batch effect is large. A low p-value (< 0.05) supports the confidence of the estimated delta value [14,15]. We found that the p-value is stable when delta is high anyway, so we only use gPCA delta in this study [16].

# Imputation accuracy

Imputation accuracy measures similarity between imputed matrix and original matrix, and is determined via the root mean square error (RMSE):

$$RMSE = \sqrt[2]{\frac{1}{n_{miss}} \sum_{i=1}^{n_{miss}} (y_i^{miss} - \dot{y}_i)^2}$$

Where $y_i^{miss}$ represents the true (but removed) data value i, and $\dot{y}_i$ is the imputed value. Since there are $n_{miss}$ missing values, the RMSE is the square root of the average of the sum of deviations between the true and imputed values. The lower the RMSE, the better the imputation accuracy.

# Imputation strategies (M1, M2 and M3)

For imputation, our design purpose is as follows: M1 represents any global imputation strategy that does not account for the batch co-variate. M2 is our proposed batch-sensitized approach. M3 is a simulated worst-case scenario where only values from the opposite batch are used for imputation. To effect M1 to

M3, we impute missing values with the global mean (M1) (i.e., mean of remaining values), same batch mean (M2 (i.e., mean of remaining values from the same batch) and opposite batch mean (M3) (i.e., mean of remaining values from opposite batch) (Fig. 1E).

# Batch simulation strategies

# Feature selection analysis

For our initial analysis, datasets used are purely simulated. The simulated dataset used for this analysis is a 20x20 matrix of normally distributed random numbers with a mean of 5 and a standard deviation (SD) of 1 (Fig. 1.1A).

Due to the specific requirement of a class factor for Harman and SVA, class effects are simulated (Fig. 1.1B). 20 samples are split evenly into 2 classes, such that all odd samples (i.e., sample 1, 3, 5, 7, 9, 11, 13, 15, 17, 19) are in class 0 and even samples (i.e., sample 2, 4, 6, 8, 10, 12, 14, 16, 18, 20) are in class 1. Class effects are then loaded onto class 0 only by multiplying class 0 by a factor of 1.5.

After adding class effects, batch effects are simulated (Fig. 1.1C). 20 samples are split evenly into 2 technical batches, such that the first 10 samples are in batch 0 and the last 10 samples are in batch 1. This class and batch allocation ensures uniform distribution of classes per batch. Mixed Batch effects (Additive + Multiplicative) are then loaded globally (i.e., all variables carry a similar component of batch-correlated effects) onto batch 0 only. This means that data from samples that belongs to batch 0, which we denote as X, will be replaced by Z(X + Y), where Y is the additive factor (arbitrarily denoted as $\sqrt{5}$) and Z is the multiplicative factor (arbitrarily denoted as 1.2). To check whether results are altered as a result of different batch effects, we repeat all analyses with additive only (i.e., X + Y) and multiplicative only (i.e., Z(X)) batch effects as well (Figure S1 – S3).

In our initial simulation, 4 Batch Effect Correction Algorithms (BECAs) (ComBat, BMC, Harman, SVA) are evaluated on batch effect removal efficacy using gPCA (Fig. 1.1F) and imputation accuracy using RMSE (Fig. 1.1G). To demonstrate reproducibility, the analysis was repeated 10 times. In our preliminary analyses, the analysis was also repeated 100 times, however, no strong differences are observed due to the relative simplicity of the simulations (Figure S4).

To demonstrate applicability to proteomics, the analysis was repeated on a proteomics dataset, RCC [17] (Fig. 1.2A). Typically, in a proteomics dataset, the rows represent the proteins and columns represent protein samples. RCC is a benchmark kidney tissue dataset comprising 3 technical replicates (batch effects are induced by combining any 2 replicates together). After omitting all rows with missing values (to measure imputation accuracy, actual data should not have any missing values), we combine the first 2 batches, each batch consisting of 4 protein samples.

To demonstrate applicability to genomics, analysis was repeated on a combined breast cancer genomics dataset, GDS4056 and GDS4057 [18] (Fig. 1.3A). Typically, in a genomics dataset, the rows represent the

genes and columns represent gene samples. GDS4056 and GDS4057 are HER-2 normal breast cancer RNA datasets from different cohorts, comprising of 2 classes: ER-positive and ER-negative subtypes (batch effects are induced by combining samples from the same subtypes but from different datasets together). To obtain a dataset with balanced batch distribution, we combine 32 ER-positive samples from GDS4056 and the first 32 ER-positive samples from GDS4057. Note that there is only one true sample class: ER-positive and 2 batches: GDS4056 and GDS4057, comprising 32 samples each.

To make our results more comparable across simulations, RCC dataset and the combined GDS4056/GDS4057 dataset was coerced to have a mean of 5 and standard deviation of 1 (values similar to our initial simulation), while maintaining the original data distribution. This is achieved by z-transforming RCC dataset by columns, and then adding 5 to all the data (Fig. 1.2A & 1.3A). We then amplify the existing batch effects of the first batch using the additive + multiplicative mixed batch effect approach (Fig. 1.2C & 1.3C).

For proteomics and genomics simulations, only 1 BECA (ComBat) is used (Fig. 1.2F & 1.3F). For the measurement of batch effect, PCA scatterplots are used on top of gPCA. RMSE is used for measurement of imputation accuracy (Fig. 1.2G & 1.3G).

# Power

To investigate the impact of batch correction on class effects, power analysis is conducted. Statistical power is the probability that a hypothesis test will find a statistically significant difference between the 2 classes when an actual difference exists [19]. Therefore, to calculate power, we first conduct a t-test for each variable (i.e., row of the data matrix) to compare the means between class 0 and class 1. The null hypothesis of the t-test is that there is no difference between the means of the 2 classes. The t-test is often interpreted by P value, defined as the probability of observing results that are equal to or more extreme than what was observed in the data, given that the null hypothesis is true [19]. If the P value is larger than the alpha level chosen (e.g., 0.05), any observed difference is assumed to be due to chance, and hence, we fail to reject the null hypothesis in this case. If P value is smaller than 0.05, we conclude that there is significant difference between the means of the 2 classes, and hence, we reject the null hypothesis [19]. After calculating the t-test P values for each variable, we calculate the power which is the number of variables with P value less than 0.05 divided over all variables.

To demonstrate reproducibility of power analysis (for initial simulation), the analysis was repeated on our genomics dataset (GDS4056/4057). To do so, we first need to simulate class effects, before amplifying the batch effects of the genomics dataset (Fig. 1.3B).

# Precision

To investigate impact on class effects, we calculate performance metrics such as Precision, Recall, False Positive Rate (FPR) and False Discovery Rate (FDR). To determine these metrics, we need to obtain the number of True Positive (TP) genes, the number of False Negative (FN) genes, the number of True Negative (TN) genes and the number of False Positive (FP) genes. To achieve this conveniently, we

modify our initial simulations such that class effects are loaded for only half of the genes (as opposed to all). The first half of the genes with simulated class effects are classified as positives while the other half with no simulated class effects are negatives (Fig. 3). Amongst positive genes, we can obtain the True Positive (TP) genes (i.e., genes with P value < 0.05) and False Negative (FN) genes (i.e., genes with P value > = 0.05). Amongst negative genes, we can obtain the True Negative (TN) genes (i.e., genes with P value > = 0.05) and False Positive (FP) genes (i.e., genes with P value < 0.05) (Fig. 3).

The formulae for Precision, Recall, FPR and FDR are as follows:

$$Precision = \frac{TP}{TP + FP} = 1 - FDR$$

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

$$FalsePositiveRate\,(FPR) = \frac{FP}{FP + TN}$$

$$FalseDiscoveryRate\,(FDR) = \frac{FP}{FP + TP}$$

# Missing value simulation

MVs are created by dropping 50% of the data randomly for each variable. This process is also referred to as missing completely at random (MCAR) (Fig. 1D).

# Simulation labels (at a glance)

To analyse the impact of MVI on imputation accuracy and global batch effect correction, we compare batch effect correction performance (via RMSE and gPCA delta scores) of BECAs on post MVI data against the original performance of BECAs on data without any missing values. This is why we have controls such as “batch” (i.e., data without missing values but with batch effects) for pre batch corrected imputed data (m1 batch, m2 batch, m3 batch) and “batch corrected” (i.e., data without missing values but with batch effects corrected by BECA) for post batch corrected imputed data (m1 batch corrected, m2 batch corrected, m3 batch corrected) (Figs. 2 & 3). However, before comparing the performance of BECAs on post MVI data versus on data without missing values, it is good to have a gauge on how good the original performance of BECAs is by comparing batch corrected against the true null (i.e., original data with class effects (if any) preserved but no batch effects). RMSE for true null is always 0 and is therefore not necessary to be shown on the RMSE graphs (Fig. 2). Even though our focus is on batch corrected imputed data, it is necessary to have pre batch corrected imputed data as control to investigate if the impact of MVI on batch effects itself matters. This analysis is then extended to power analysis to investigate the impact on class effects after batch effect correction, which is why the controls and legends for power and t-statistic remain the same as gPCA delta. The label “base power” in the power graphs just refers to the power of the true null (Fig. 4).

## Declarations

# Data availability

The breast cancer genomics datasets analysed during the current study are available in GEO, [IDs: GDS4056 and GDS4057]. The RCC proteomics dataset is available in PRIDE via accession PXD000672.

# Author contributions

PYQS implemented analyses, performed initial drafting of the manuscript, and development of figures. WWBG conceptualized, supervised, provided critical feedback and wrote the manuscript.

# Declaration of interests

The authors declare no competing interests.

# Acknowledgments

This work is supported by the Ministry of Education, Singapore, under its Academic Research Fund Tier-1 (RG35/20) to WWBG.

## References

1 Aittokallio, T. Dealing with missing values in large-scale studies: microarray data imputation and beyond. *Briefings in Bioinformatics***11**, 253-264, doi:10.1093/bib/bbp059 (2009).

2 Goh, W. W. B., Wang, W. & Wong, L. Why batch effects matter in omics data, and how to avoid them. *Trends in biotechnology***35**, 498-507 (2017).

3 Luo, J. *et al.* A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *The pharmacogenomics journal***10**, 278-291 (2010).

4 Kupfer, P. *et al.* Batch correction of microarray data substantially improves the identification of genes differentially expressed in rheumatoid arthritis and osteoarthritis. *BMC medical genomics***5**, 23 (2012).

5 Konstantinopoulos, P. A. *et al.* Integrated analysis of multiple microarray datasets identifies a reproducible survival predictor in ovarian cancer. *PloS one***6**, e18202 (2011).

6 Chen, C. *et al.* Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PloS one***6**, e17238 (2011).

7 Webb-Robertson, B.-J. M. *et al.* Review, evaluation, and discussion of the challenges of missing value imputation for mass spectrometry-based label-free global proteomics. *Journal of proteome research***14**, 1993-2001 (2015).

8 Belorkar, A. & Wong, L. GFS: fuzzy preprocessing for effective gene expression analysis. *BMC bioinformatics***17**, 540 (2016).

9 Zhao, Y., Wong, L. & Goh, W. W. B. How to do quantile normalization correctly for gene expression data analyses. *Scientific reports***10**, 1-11 (2020).

10 Liu, M. & Dongre, A. Proper imputation of missing values in proteomics datasets for differential expression analysis. *Briefings in Bioinformatics*, doi:10.1093/bib/bbaa112 (2020).

11 Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics***8**, 118-127, doi:10.1093/biostatistics/kxj037 (2006).

12 Leek, J. T. & Storey, J. D. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet***3**, e161 (2007).

13 Oytam, Y. *et al.* Risk-conscious correction of batch effects: maximising information extraction from high-throughput genomic datasets. *BMC bioinformatics***17**, 332 (2016).

14 Nygaard, V., Rødland, E. A. & Hovig, E. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics***17**, 29-39, doi:10.1093/biostatistics/kxv027 (2015).

15 Reese, S. E. *et al.* A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. *Bioinformatics***29**, 2877-2883, doi:10.1093/bioinformatics/btt480 (2013).

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

17 Goh, W. W. B. & Wong, L. Advancing clinical proteomics via analysis based on biological complexes: A tale of five paradigms. *Journal of proteome research***15**, 3167-3179 (2016).

18 Iwamoto, T. *et al.* Gene pathways associated with prognosis and chemotherapy sensitivity in molecular subtypes of breast cancer. *Journal of the National Cancer Institute***103**, 264-272 (2011).

19 Sullivan, G. M. & Feinn, R. Using effect size—or why the P value is not enough. *Journal of graduate medical education***4**, 279-282 (2012).

# Figures

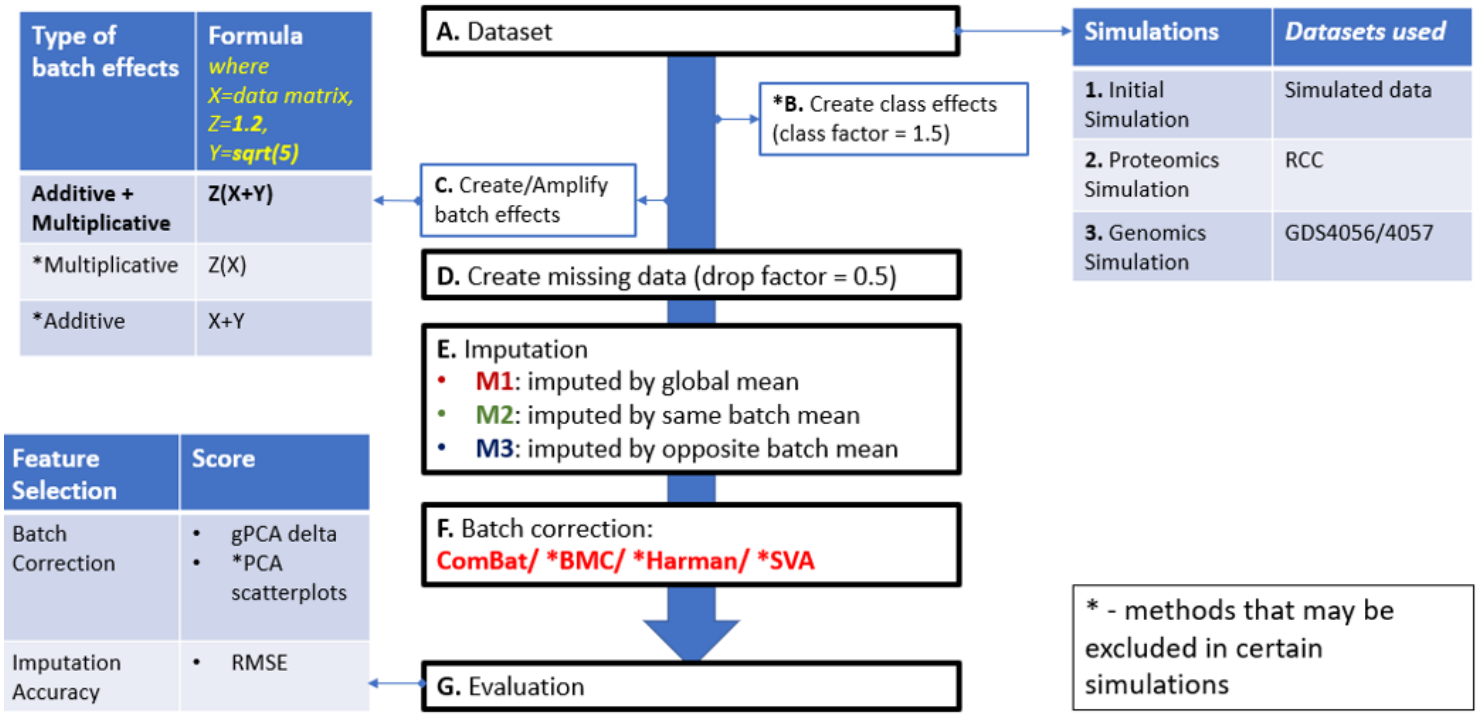
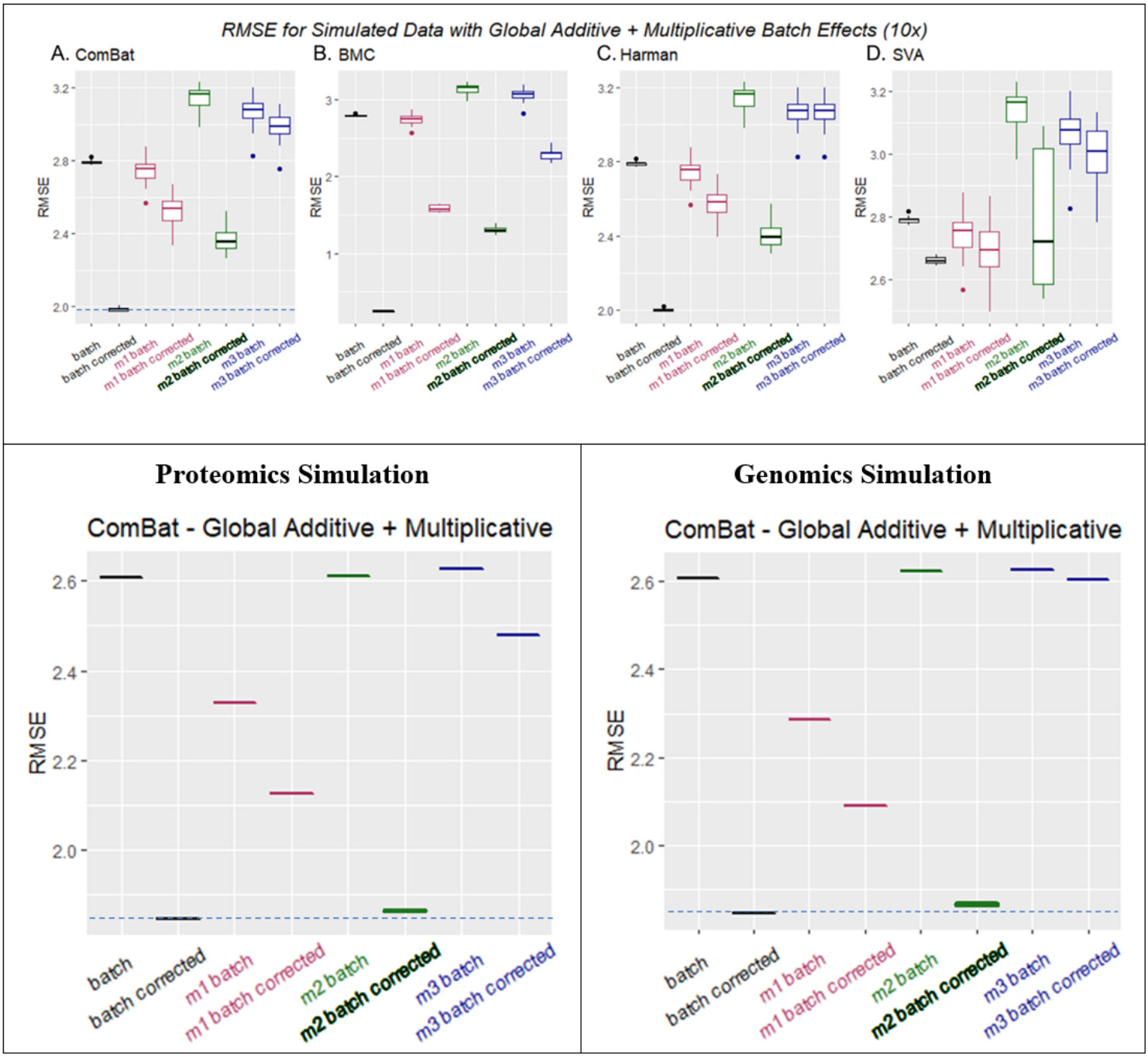


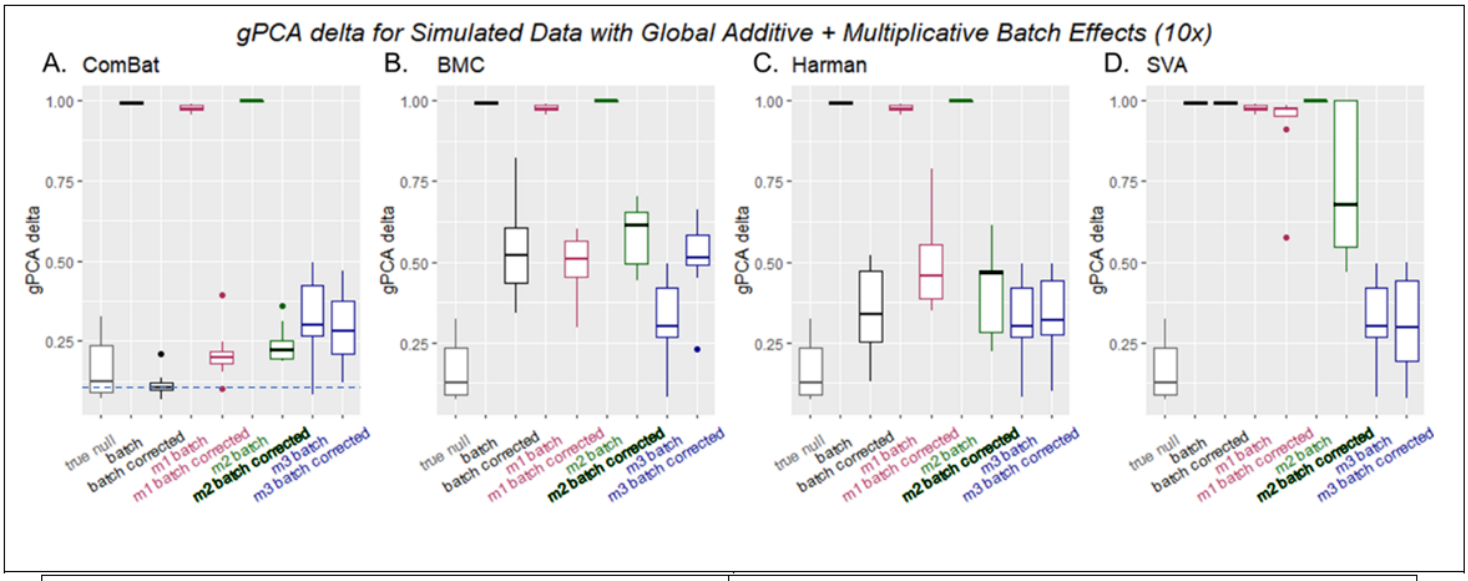

**Figure 1**

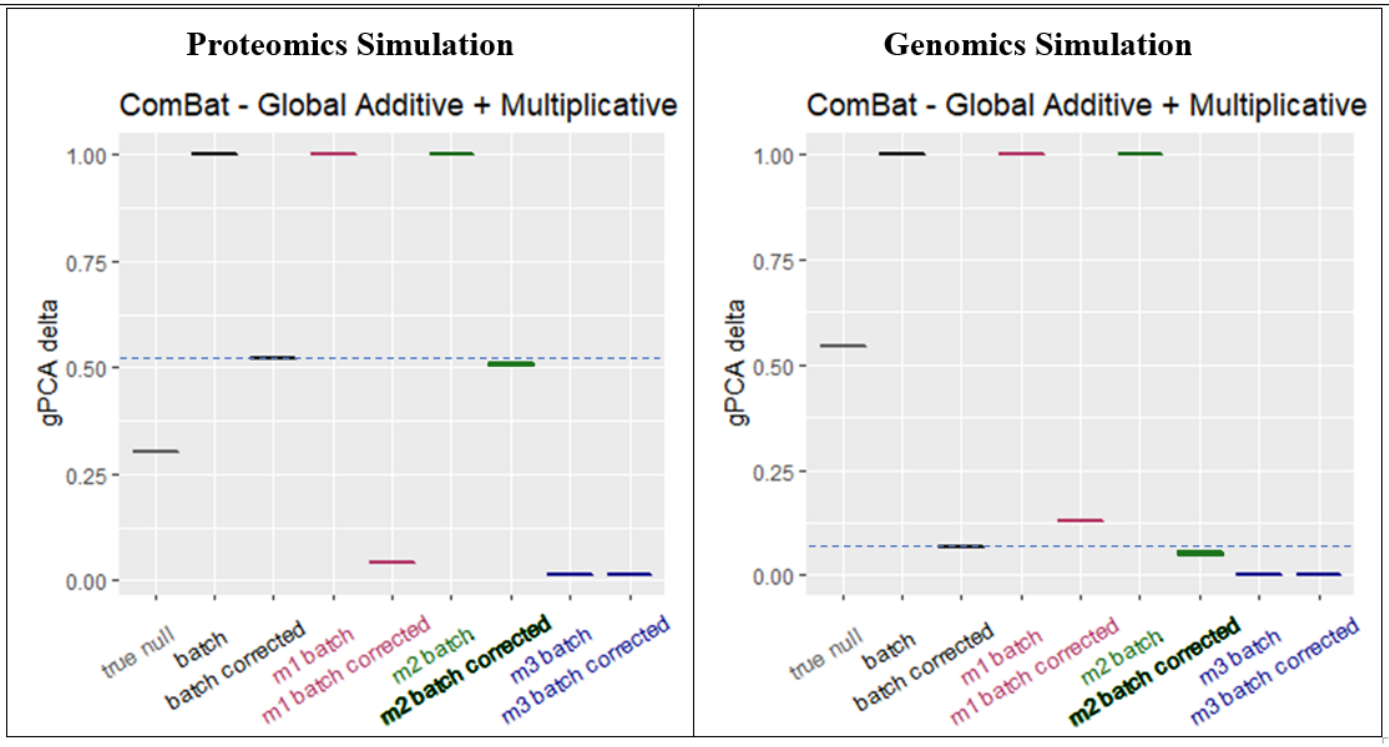
**A.** 3 types of datasets are used for this analytical pipeline: simulated data for **1.** Initial Simulation, Renal Control dataset (RCC) for **2.** Proteomics Simulation and lastly, GDS4056/4057 combined dataset for **3.** Genomics Simulation. Analytical pipeline consists of **B.** simulating class and **C.** batch effects, followed by **D.** introduction of missing values, **E.** imputation, **F.** batch correction and **G.** evaluation. (see **Supplementary Methods** for more detailed captions for each simulation).




## Figure 2

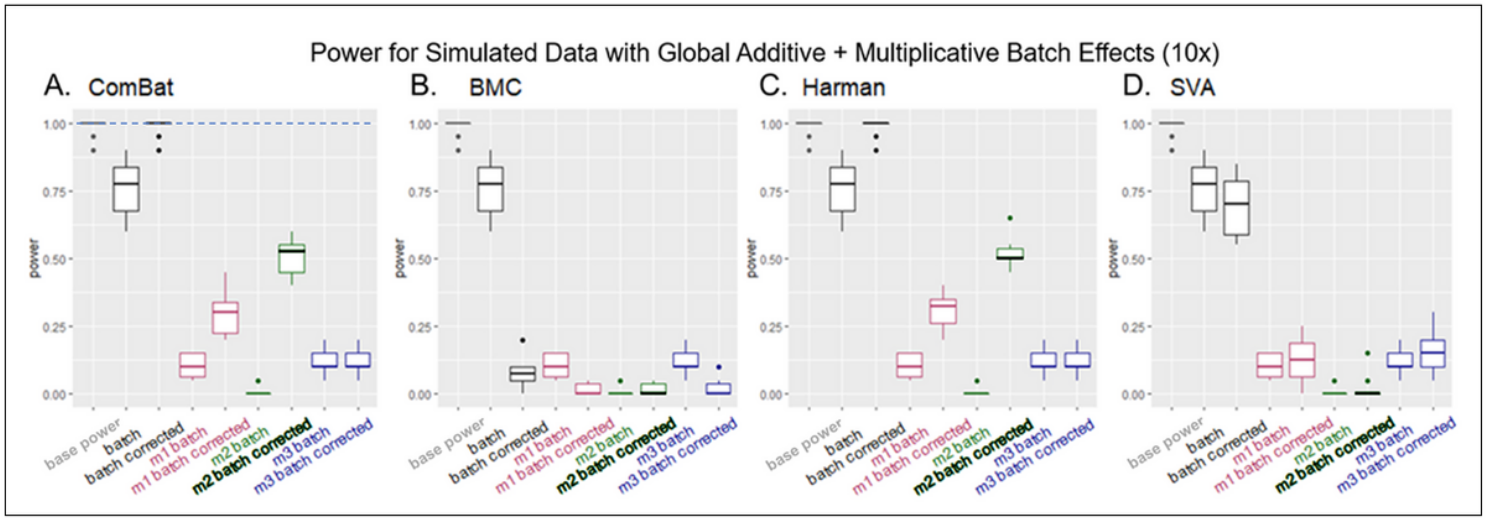
4 Batch effect correction algorithms (BECAs) are used for evaluation of imputation accuracy for **1.** Initial Simulation based on the root mean square error (RMSE): **A.** ComBat **B.** BMC **C.** Harman **D.** SVA. Only ComBat is used for subsequent evaluation of imputation accuracy for **2.** Proteomics Simulation and **3.** Genomics Simulation based on the root mean square error (RMSE). Lower values indicate higher similarity to original data matrix.



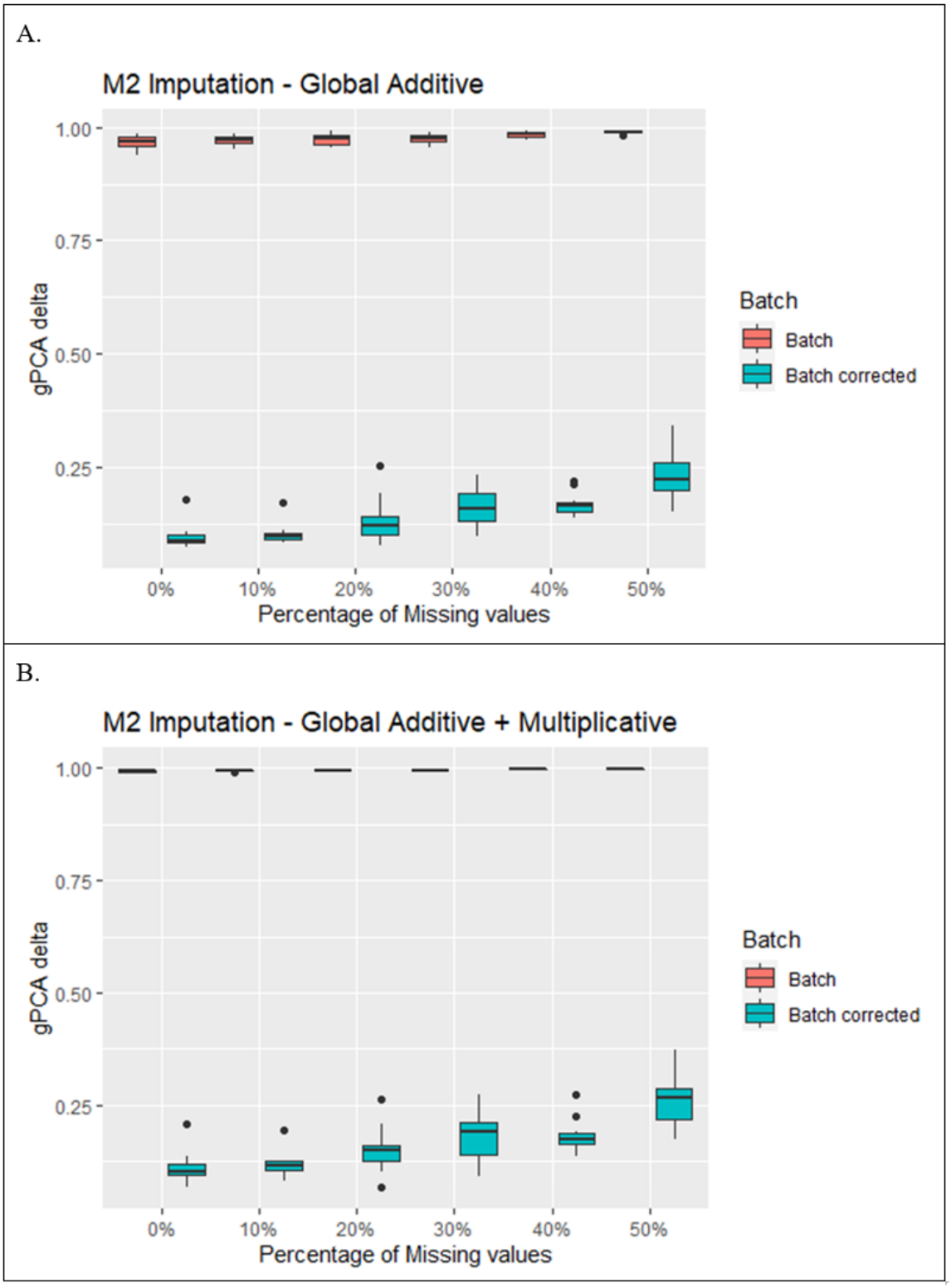

## Figure 3

4 Batch effect correction algorithms (BECAs) are used for evaluation of batch correction for **1.** Initial Simulation based on the gPCA delta: **A.** ComBat **B.** BMC **C.** Harman **D.** SVA. Only ComBat is used for evaluation of batch correction for **2.** Proteomics Simulation and **3.** Genomics Simulation based on the gPCA delta. Lower values indicate less batch-correlated separation in data.



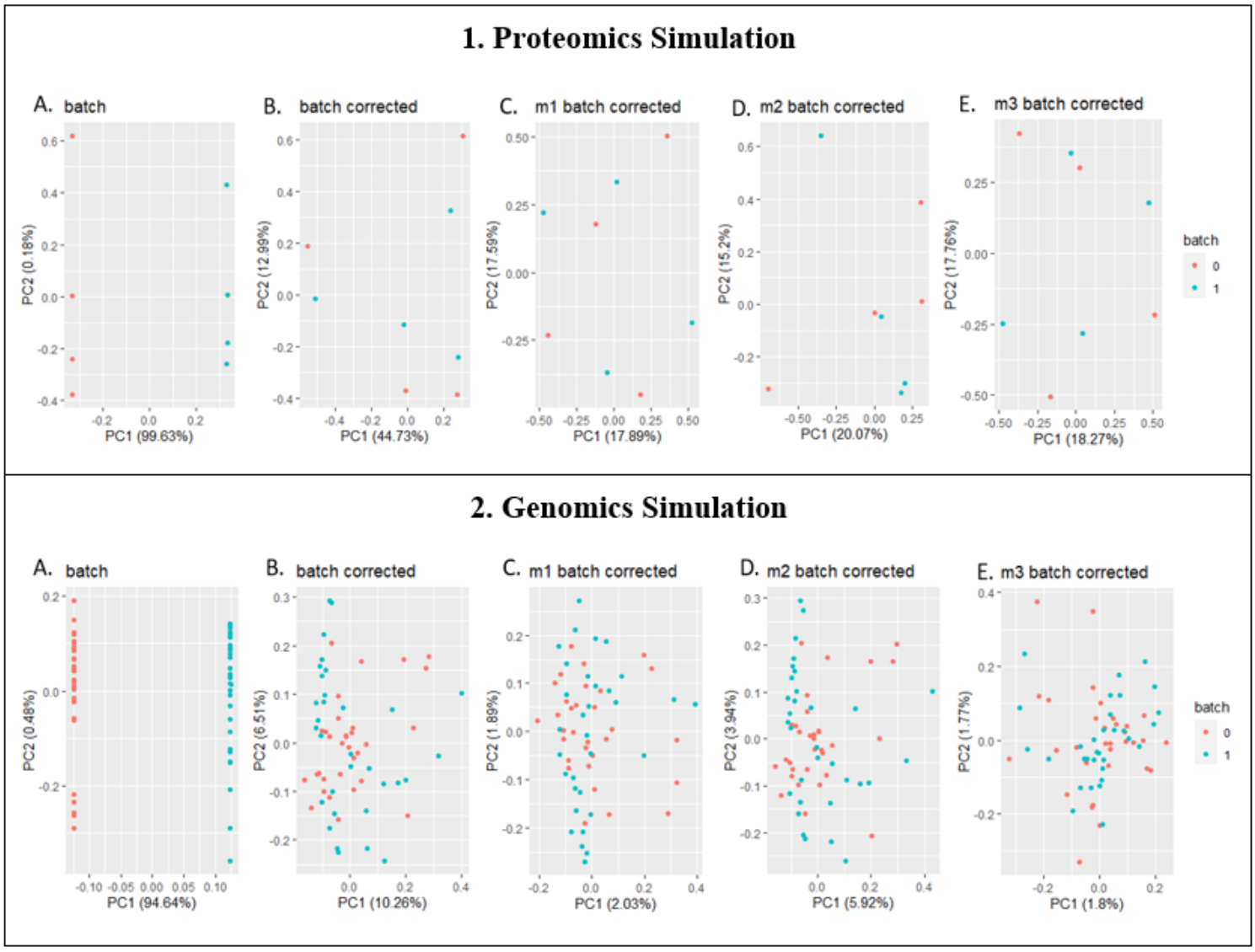

## Figure 4

4 Batch effect correction algorithms (BECAs) are used for evaluation of power for Initial Simulation based on statistical feature selection : **A.** ComBat **B.** BMC **C.** Harman **D.** SVA. Higher values indicate better performance (higher recall of correct features).



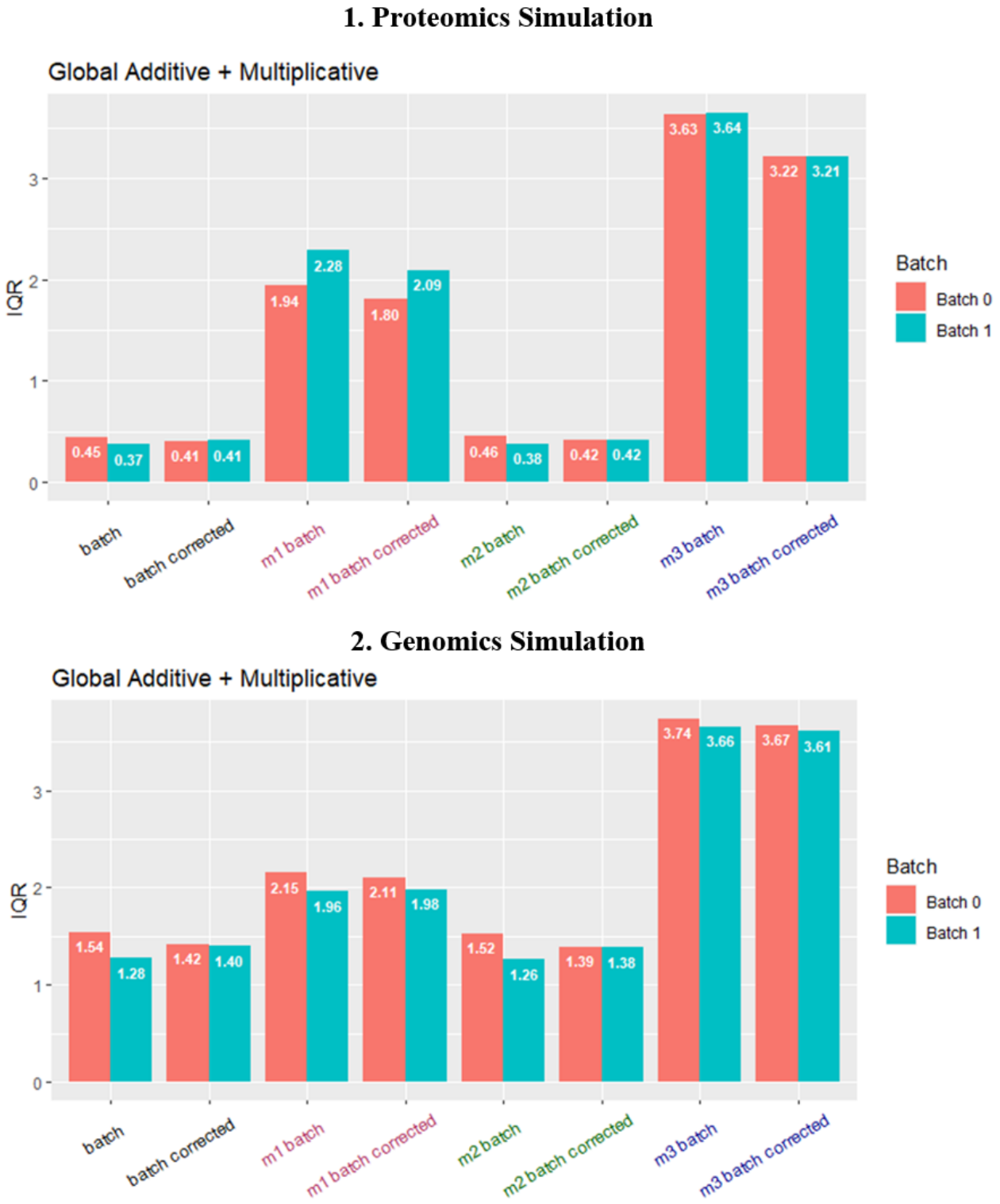

## Figure 5

Initial simulation, with varying % of missing values (i.e., 10%, 20%, 30%, 40%, 50%) imputed by M2 only, is used for evaluation of batch correction for M2 based on gPCA delta. Only ComBat is used as BECA for this evaluation. gPCA delta results for both **A.** additive only and **B.** mixed batch effects (Additive + Multiplicative) scenarios showed that after batch correction, remnant batch in M2 is attenuated given lower % missing values.



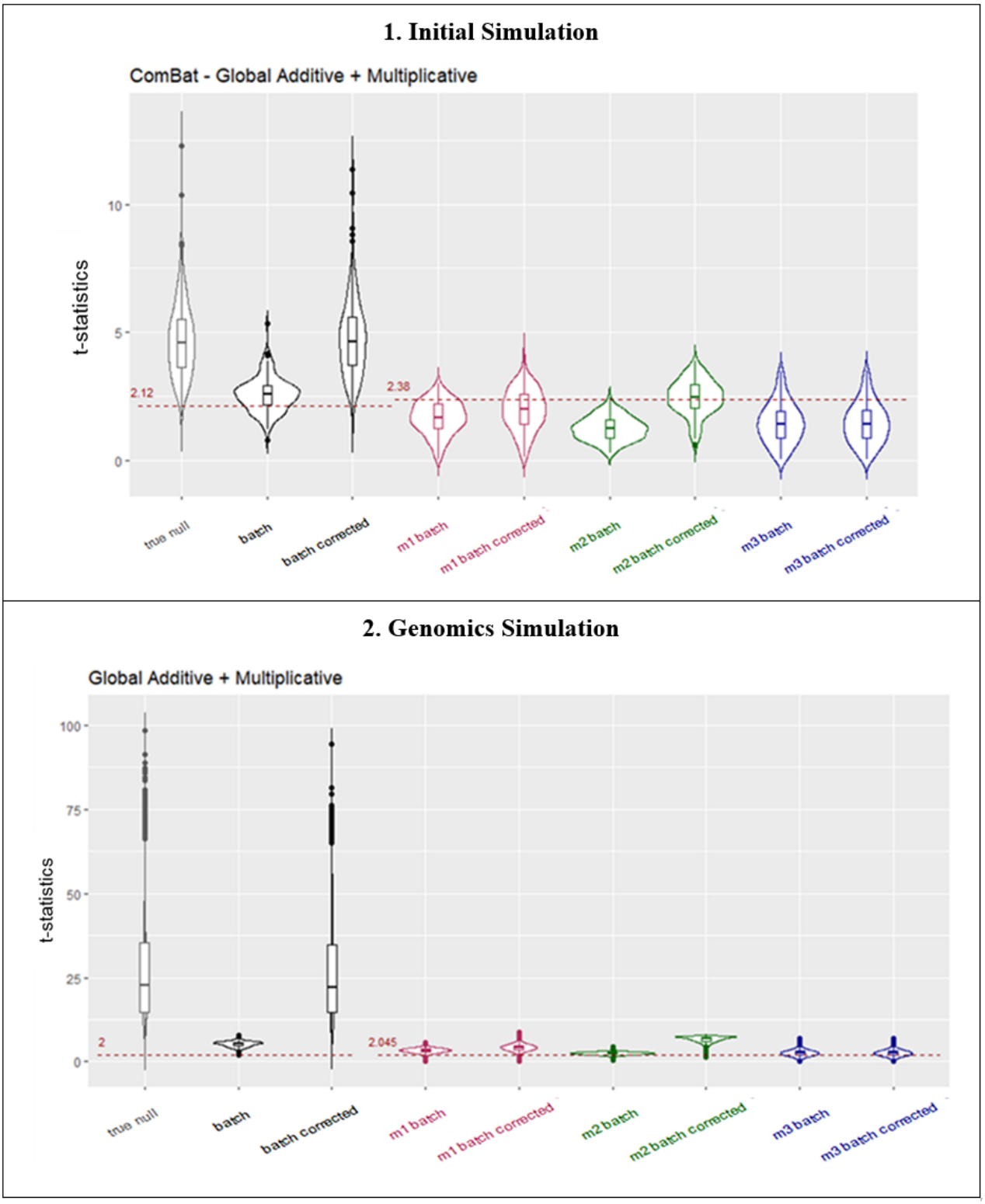

## Figure 6

PCA Scatterplots for **1.** Proteomics Simulation and **2.** Genomics Simulation showed that despite reporting higher gPCA levels for M2, samples appear well-mixed, with no apparent batch effects for all imputation strategies (M1 to M3), given the first two principal components (PC1 and PC2) (c.f. Figure S5 for full version)




## Figure 7

Interquartile Range (IQR) for **1.** Proteomics Simulation and **2.** Genomics Simulation showed that while M2 preserved similar sample variances to original data, M1 and M3 variances were grossly inflated.




## Figure 8

t-statistic distributions for **1.** Initial Simulation and **2.** Genomics Simulation reveal that although power is the best for M2, all imputation strategies (M1 to M3) suffer from a reduction in effect size.


**Figure 9**

Performance Metrics for Genomics Simulation reveal that M2 performs the best among post MVI data. Higher values indicate better performance: **A.** Precision **B.** Recall. Lower values indicate better performance: **C.** False Positive Rate (FPR) **D.** False Discovery Rate (FDR).

# Supplementary Files

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

- Supplementarymaterials.docx