**Why batch effect sensitization is important for missing value imputation**

Weijia Kong1^, Priscila Yun Qian Sun1, Limsoon Wong2\*, Wilson Wen Bin Goh1\*

1 School of Biological Sciences, Nanyang Technological University, Singapore 637551, Singapore

2 Department of Computer Science, National University of Singapore, Singapore 117417, Singapore

^ First author; \* Corresponding author(s).

E-mail: [wilsongoh@ntu.edu.sg](mailto:wilsongoh@ntu.edu.sg) (Goh WWB); [wongls@comp.nus.edu.sg](mailto:wongls@comp.nus.edu.sg) (Wong L)

**Running title**: Kong et al. */ Why batch effect sensitization is important for missing value imputation*

# Supplementary materials and methods

## 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 [1].

### 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 [2].

### 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 [3].

### 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 [4].

## 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 [5].

### 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 YTX, 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 [4,5]. We found that the p-value is stable when delta is high anyway, so we only use gPCA delta in this study [6].

## Imputation accuracy

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

Where represents the true (but removed) data value i, and is the imputed value. Since there are nmiss 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).

## 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.

Due to the specific requirement of a class factor for Harman and SVA, class effects are simulated (Figure 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. 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 ) 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.

In our initial simulation, 4 Batch Effect Correction Algorithms (BECAs) (ComBat, BMC, Harman, SVA) are evaluated on batch effect removal efficacy using gPCA and imputation accuracy using RMSE. 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 (data not shown).

To demonstrate applicability to proteomics, the analysis was repeated on a proteomics dataset, RCC [7]. 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 [8]. 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. We then amplify the existing batch effects of the first batch using the additive + multiplicative mixed batch effect approach.

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

### Feature selection

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 (Figure 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) (Figure 3).

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

## 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).

# Supplementary results

Graphical user interface, text

Description automatically generated

Supplementary 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.

|  |
| --- |
| A.  **Chart  Description automatically generated** |
| B.  **Chart, box and whisker chart  Description automatically generated** |

Supplementary Figure 2 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.

|  |
| --- |
| **1. Proteomics Simulation**  Chart  Description automatically generated with low confidence |
| **2. Genomics Simulation**  Chart, scatter chart  Description automatically generated |

Supplementary Figure 3 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)

|  |  |
| --- | --- |
| A.  Chart, scatter chart  Description automatically generated | B.  Chart  Description automatically generated |

Supplementary Figure 4 A. Power and B. Recall for reduced genomics data (20x20 matrix) show that after reducing sample size of genomics data, M2 batch corrected no longer performs as well as batch corrected.

References

**1** Johnson, W.E. et al. (2006) Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 8 (1), 118-127

**2** Leek, J.T. and Storey, J.D. (2007) Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet* 3 (9), e161

**3** Oytam, Y. et al. (2016) Risk-conscious correction of batch effects: maximising information extraction from high-throughput genomic datasets. *BMC bioinformatics* 17 (1), 332

**4** Nygaard, V. et al. (2015) Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics* 17 (1), 29-39

**5** Reese, S.E. et al. (2013) A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis. *Bioinformatics* 29 (22), 2877-2883

**6** Zhou, L. et al. (2019) Examining the practical limits of batch effect-correction algorithms: When should you care about batch effects? *Journal of Genetics and Genomics* 46 (9), 433-443

**7** Goh, W.W.B. and Wong, L. (2016) Advancing clinical proteomics via analysis based on biological complexes: A tale of five paradigms. *Journal of proteome research* 15 (9), 3167-3179

**8** Iwamoto, T. et al. (2011) Gene pathways associated with prognosis and chemotherapy sensitivity in molecular subtypes of breast cancer. *Journal of the National Cancer Institute* 103 (3), 264-272

**9** Sullivan, G.M. and Feinn, R. (2012) Using effect size—or why the P value is not enough. *Journal of graduate medical education* 4 (3), 279-282