

Harmonizing Data Variability: A Comparative Analysis of Batch Effect Removal Methods via Synthetic Batch Effect Introduction

Alaa Bakry^{1,*}, Andreas Eberhard², Debasmita Asari³, Chou Yi Hsu⁴, Shahid Ullah⁵, Lilianne Dominguez Céspedes⁶

^{1,2,3}Center for Bioinformatics, Saarland University, Saarbrücken, Saarland, Germany.

⁴Department of Pharmacy, Chia Nan University of Pharmacy and Science, Tainan City, Taiwan.

⁵Department of Bioinformatics and Biochemistry, S. Khan Lab, Mardan, Khyber Pakhtunkhwa, Pakistan.

⁶Department of Clinical Bioanalysis, Filial of Medical Sciences of Holguín, Avenida Lenin, Esquina Aguilera, Holguín, Cuba.

abakry@zewailcity.edu.eg¹, andreas.eberhard@bioinformatik.uni-saarland.de², debasmita.asari@bioinformatik.uni-saarland.de³, joyhsu@yuanangroup.com.tw⁴, shahid@habdsk.org⁵, lilianned40@gmail.com⁶

Abstract: Batch effects (BEs) systematic, non-biological discrepancies introduced when samples are processed across different locations, platforms, or times remain a persistent obstacle to the integrative analysis of RNA-Seq datasets. Reliable correction methods must remove technical variation without eroding genuine biological signal, yet a standardised way to benchmark these methods against data with known, controllable BE magnitude is lacking. Researchers introduced synthetic batch effects into two public RNA-Seq datasets using three complementary strategies additive random noise scaled to a fraction of the standard deviation, row/column mean augmentation, and cluster-mean augmentation following k-means partitioning and then applied four established BE-removal methods (ComBat, Surrogate Variable Analysis (SVA), BEclear, and Mutual Nearest Neighbors (MNN)) to recover the original signal. All four methods reduced batch-associated variance relative to the uncorrected data. MNN achieved the strongest correction, yielding the lowest post-correction correlation between principal components (-1.316×10^{-5}). At the same time, ComBat and SVA showed comparable, moderate performance, and BEclear performed well but at higher computational cost. Researchers found no existing software tool purpose-built for injecting synthetic batch effects, motivating our three custom introduction strategies.

Keywords: Non-Biological Discrepancies; Persistent Obstacle; Integrative Analysis; RNA-Seq Datasets; Eroding Genuine Biological Signal; Surrogate Variable Analysis; Mutual Nearest Neighbours.

Received on: 27/09/2024, **Revised on:** 22/12/2024, **Accepted on:** 27/04/2025, **Published on:** 15/09/2026

Journal Homepage: <https://www.fmdbpublish.com/user/journals/details/FTSASS>

DOI: <https://doi.org/10.69888/FTSASS.2026.000768>

Cite as: A. Bakry, A. Eberhard, D. Asari, C. Y. Hsu, S. Ullah, and L. D. Céspedes, “Harmonizing Data Variability: A Comparative Analysis of Batch Effect Removal Methods via Synthetic Batch Effect Introduction,” *FMDB Transactions on Sustainable Applied Sciences*, vol. 3, no. 2, pp. 62–73, 2026.

Copyright © 2026 A. Bakry *et al.*, licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under [CC BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/), which permits copying, remixing, redistributing, transformation, and building upon the material in any medium so long as the original work is properly cited.

1. Introduction

High-throughput gene expression profiling by RNA sequencing (RNA-Seq) is routinely confounded by batch effects (BEs): systematic technical variation arising from processing samples in different laboratories, at different times, on different

*Corresponding author.

platforms, or by different personnel [2]. Unlike random noise, BEs are structured and reproducible within a batch [1]. Unlike bias, they are not necessarily confounded with a specific biological group—yet, if left uncorrected, they can dominate genuine biological signal and distort downstream differential expression, clustering, and classification analyses [3]. Two broad strategies for combining data from multiple sources have been described: integrative analysis, in which datasets are merged before analysis, and meta-analysis, in which per-dataset results are combined post hoc [4]. Integrative analysis offers greater statistical power but requires adequate removal of BEs beforehand. Existing correction methods fall into two principal families: location-scale (LS) methods, which adjust each batch's mean and/or variance, and matrix-factorization (MF) methods, which decompose the expression matrix to isolate batch-associated components [5].

Distance-based approaches, such as Mutual Nearest Neighbors (MNN), form another increasingly popular family, particularly for single-cell and bulk RNA-Seq integration [6]. A recurring difficulty in this field is the lack of ground truth: in real datasets, the true BE-free signal is unknown, making it difficult to benchmark correction methods objectively [7]. Even recently, the authors of reComBat noted that residual unwanted variation “currently cannot be solved by any common batch correction algorithm”, underscoring the need for controlled evaluation frameworks [8]. Researchers reasoned that batch effects of known magnitude and structure could be synthetically introduced into otherwise clean RNA-Seq datasets, providing a controllable benchmark against which correction methods could be evaluated without confounding by unknown pre-existing technical variation. In this study, researchers therefore (i) surveyed available tools for synthetic BE introduction and, finding none directly applicable, developed three complementary strategies for injecting BEs of controllable magnitude into RNA-Seq count data; (ii) applied four representative correction methods, ComBat, SVA, BEclear, and MNN, to the resulting datasets; and (iii) compared their ability to recover the original, BE-free structure of the data using principal component analysis and correlation-based metrics [9].

1.1. Definition and Sources of Batch Effects

A batch effect is variability introduced into the resulting data during sample handling, arising from disparities in procedures, environmental conditions, or the instrumentation used across different laboratories; it can also occur within a single laboratory across different processing runs [10]. Batch effects should be distinguished conceptually from two related but distinct sources of unwanted variation: noise, which refers to non-systematic fluctuations caused by unaccounted sources of variation, and bias, which encompasses technical and confounding factors that inadvertently link specific sample characteristics to experimental groups [11]. Unlike noise, batch effects are systematic and reproducible within a batch; unlike bias, they are not necessarily confounded with the biological variable of interest, although in poorly designed experiments the two can become entangled. The primary contributors to batch effects fall into three broad categories [12]. Environmental factors include seasonal variation, humidity, ambient temperature, and the processing site's geographical location [13].

Laboratory conditions encompass procedural differences between operators, differences in array or library handling, sample storage conditions, washing or wash-step protocols, and variability between reagent lots or chip/flow-cell types [14]. Other subtler contributors include uneven distribution of surface coatings or fluids on processing consumables, manufacturing imperfections in glassware or plasticware, ambient dust particles, and even the specific physical location within a facility where an experiment is conducted [15]. Because these sources of variation are so numerous and often uncontrollable, experimental design alone generally cannot prevent batch effects entirely; instead, computational correction after data generation is required before cross-batch or cross-study comparisons can be made reliably [16]. Even so, it is important to note that no available correction method is guaranteed to remove unwanted variation completely — as the authors of reComBat explicitly caution, this residual-variation problem “currently cannot be solved by any common batch correction algorithm” [17].

1.2. Categories of Batch Effect Removal Methods

Batch effect removal strategies are conventionally grouped into two principal families, with a smaller number of additional, less widely used approaches [19]. Matrix-factorization (MF) methods treat the largest source of variance in the gene-by-sample expression matrix as batch-related rather than the biological variable of interest [18]. These methods first factorize the matrix—typically via singular value decomposition (SVD) or principal component analysis (PCA)—and then apply a subsequent step, such as Surrogate Variable Analysis (SVA) or Remove Unwanted Variation in two steps (RUV-2), to more precisely isolate and remove batch-associated components [20]. MF methods rest on two key assumptions: first, that the expression matrix can be adequately approximated by a small number of rank-one components recoverable through factorisation; and second, that the biological variation of interest is statistically independent of the batch-associated variation [21]. RUV-2, in particular, relies on a defined set of negative-control genes assumed to be uncorrelated with the biological factor of interest, so that any variation observed in these control genes can be attributed entirely to unwanted, non-biological sources; RUV-2 combines a simple two-step factor-analysis-then-regression procedure and has been reported to outperform both empirical Bayes methods and ordinary regression in several settings [22]. Location-scale (LS) based methods, by contrast, assume that the origin of the batch effect

can be explicitly modeled in terms of its effects on the location (mean), scale (variance), or the full distribution of the data within each batch [23].

They adjust batches to conform to a common model without deliberately removing the underlying biological signal [24]. This family includes ComBat, which researchers describe in detail below; Distance-Weighted Discrimination (DWD), a linear discrimination method that treats batch membership as a classification label, identifies an optimal hyperplane maximally separating batches, and shifts each batch's mean toward that hyperplane while considering all data points (not merely support vectors) in defining the boundary Cross-Platform Normalization (XPN), which employs a block-linear model to segment the expression matrix into homogeneous gene-and-sample blocks for normalization across platforms; ratio-based methods, which normalize each sample against a reference array constructed from averaged reference or negative-control samples; gene standardization; and simple batch mean-centering [25]. A smaller category of discretization-based methods (e.g., quantile discretization, the fRMA barcode) and other miscellaneous approaches (e.g., median rank scores, quantile normalization, gene quantiles) also exist but are used less commonly and are not evaluated further here [26].

An important distinction among LS methods concerns whether they operate gene-by-gene or account for inter-gene relationships. Approaches such as simple mean-centring and DWD generally assume that batch bias produces only a shift in each gene's mean [27]. At the same time, DWD considers inter-gene relationships when defining its separating hyperplane; the resulting adjustment remains parallel to the coordinate axes and therefore does not modify the variance–covariance structure among genes. This is a non-trivial limitation, since batch effects have been observed in practice to exhibit multivariate character, altering the variance–covariance structure of genes rather than shifting each gene independently [28]. Methods that explicitly consider covariance structure, including centroid alignment and certain multi-batch covariance-adjustment (MBCA) approaches, can, in principle, better capture these effects at the cost of additional model complexity [29]. Finally, the choice between LS and MF approaches—and among specific methods within each family—generally depends on whether the source of batch bias is known. DWD, empirical Bayes (ComBat), and XPN are best suited to situations where the technical source of variation is identifiable and can be explicitly modeled, whereas RUV-2 and SVA are more appropriate when the source of batch bias is unknown and must be inferred statistically from the data.

2. Materials and Methods

2.1. Datasets

Researchers obtained two publicly available RNA-Seq datasets from the NCBI Gene Expression Omnibus (GEO). The first, GSE213730 ('Tspan7'), comprises human samples examining the effect of tetraspanin-7 knockdown on pancreatic beta-cell exocytosis, with a scrambled short hairpin RNA (shRNA) control group (SCRAM) and a Tspan7-targeting shRNA knockdown group (ShTspan7), each with three replicates. The second, GSE173078 ('gingival'), profiles mRNA expression in human gingival tissue across three periodontal states: periodontal health, gingivitis, and periodontitis. Both datasets were pre-normalised by their original depositors; no additional normalisation (e.g., counts-per-million or log transformation) or missing-value imputation was required or applied.

2.1.1. Data Collection, Preprocessing, and Evaluation Metrics

Researchers retrieved raw gene expression counts for both datasets directly from GEO and screened them for batch effects before any synthetic manipulation. Preprocessing consisted of removing the sample- and gene-identifier columns from the raw count matrices, leaving only numeric expression values; the resulting expression matrix contained gene-level values for each sample in each dataset. As noted above, because both source datasets were already normalized by their depositors, researchers applied no further counts-per-million, transcripts-per-million, or log-transformation steps, and researchers encountered no missing values at any stage of the analysis. To evaluate and compare the four correction methods after introducing synthetic batch effects, researchers used several complementary evaluation approaches. Researchers applied principal component analysis before and after each correction to visualize whether batch-associated structure had been reduced. K-means clustering was used as a secondary check, on the reasoning that samples clustering primarily by treatment or biological condition, rather than by batch, is consistent with successful batch effect removal; conversely, a dataset in which clustering is dominated by batch membership, or in which surrogate variable analysis identifies at least one strong surrogate variable, is indicative of a residual, uncorrected batch effect.

Outlier detection, using the R 'Routliers' package, was additionally applied to assess whether correction procedures introduced implausible or extreme values not present in the original data. Together, these metrics were intended to capture both the reduction of technical (batch-associated) variability and the preservation of genuine biological variability specific to each sample or condition, rather than relying on a single summary statistic. Researchers performed the full analysis in R version 4.0.5 (2021-03-31), used Python (PyCharm 3.8.8) for auxiliary scripting, and sourced all datasets from the freely available

NCBI repository. In addition to the four methods evaluated in depth here, the authors reviewed related implementations, including ComBat-seq, a count-based extension of ComBat, and SVA-seq. Researchers conceptually compared their underlying algorithms with ComBat, BEclear, DWD, and SVA, which informed the selection of the four representative methods ultimately applied and benchmarked in full.

2.2. Synthetic Batch Effect Introduction

Researchers did not identify any existing software that directly supports the controlled injection of synthetic batch effects into RNA-Seq count matrices. Researchers therefore implemented three strategies:

- **Additive random noise:** Researchers added random values drawn from the range $[1, 2 \times \text{SD}]$ (SD = the dataset's standard deviation) to expression values, simulating BEs in a controlled, systematic manner.
- **Row/column mean augmentation:** Researchers added the row mean or column mean to each corresponding value, introducing an additional layer of structured variability intended to emulate real-world batch scenarios.
- **Cluster-mean augmentation:** Researchers partitioned the dataset via k-means clustering (optimal cluster number determined by the elbow method on within-cluster sum of squares, WCSS) into eight clusters; researchers discarded one singleton cluster as uninformative and added the mean of each of the remaining seven clusters to every data point within that cluster.

Researchers also tested a fourth exploratory approach—introducing BEs after first removing statistical outliers—but did not pursue it. This strategy first applied outlier detection and removal as an initial correction step, then introduced synthetic biological effects on the cleaned matrix (Figure 1).

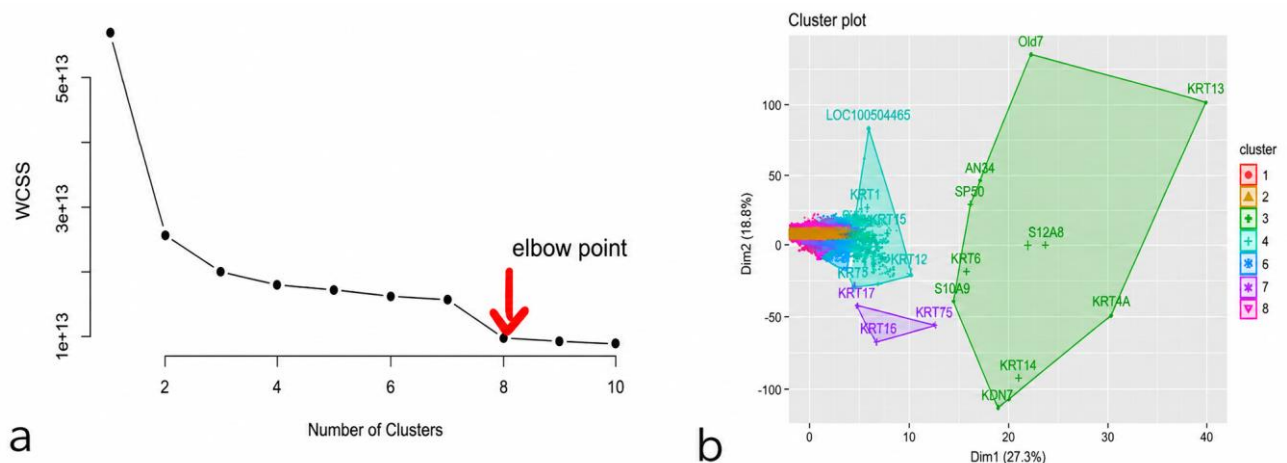


Figure 1: Determination of cluster number for the cluster-mean augmentation strategy. (a) Elbow plot of within-cluster sum of squares (WCSS) versus number of clusters, identifying eight clusters as optimal. (b) Resulting cluster partition of the dataset, later reduced to seven clusters after exclusion of one uninformative singleton cluster

Contrary to expectation, this sequencing produced a substantial proportion of newly flagged extreme values: using the median absolute deviation (MAD)-based outlier detection implemented in the Routliers package, the acceptable range of values was estimated at approximately $[-258, 344]$ ($\text{MAD} \approx 43$), and more than 290,000 individual entries — over 50% of the gingival dataset fell outside this range and were flagged as extremely high outliers after the synthetic effect was added. Because this method destabilized rather than clarified the dataset's structure, researchers excluded it from the four introduction strategies carried forward for formal benchmarking. However, it remains informative as an illustration of how sensitive outlier-based preprocessing can be to the order in which cleaning and effect-injection steps are applied.

2.3. Batch Effect Correction Methods

Researchers applied four established BE-removal methods, spanning the location-scale, matrix-factorization, and distance-based families, to the datasets after introducing synthetic BE. ComBat, implemented via the sva R package, models the expression value of gene g in sample j of batch i as $Y_{ijg} = \alpha_g + X\beta_g + \gamma_{ig} + \delta_{ig}\epsilon_{ijg}$, where α_g is the overall gene expression level, and X is the design matrix of biological covariates. γ_{ig} and δ_{ig} represent additive and multiplicative batch effects, respectively. Batch-specific parameters are estimated via linear modeling, shrunk toward the overall mean using empirical Bayes, and then

subtracted to yield batch-adjusted data Y^*_{ijg} . Surrogate Variable Analysis (SVA-seq; Leek and Storey, [12]) extends the model $Y = X\beta + ZB + \epsilon$ by estimating surrogate variables S via singular value decomposition of residuals from an initial model fit, yielding $Y = X\beta + ZB + S + \epsilon$. SVA-seq adapts this framework to RNA-Seq count data using negative binomial modeling rather than the Gaussian assumptions of standard SVA. The Tspan7 dataset, containing two experimental groups, was modeled with a single surrogate variable, whereas the more heterogeneous gingival dataset required four.

BEclear, implemented via the BEclear R package with parallelization via the snowfall package, treats the $m \times n$ expression matrix D as approximable by the product of two lower-rank matrices L ($m \times r$) and R ($r \times n$) for all BE-unaffected entries, estimated via gradient descent to minimize the global loss between $[LR]$ and D . BE-affected entries, identified via a weighted scoring scheme based on median differences across batches, are selectively replaced by their corresponding entries in the completed matrix $D_{comp} = [LR]$, with predicted values clipped to the valid range. Mutual Nearest Neighbors (fastMNN; implemented via the batchelor/mnncorrect R packages) identifies, for each sample in one batch, its nearest neighbor in another batch (and vice versa), retaining only mutual nearest-neighbor pairs. A correction vector is computed as the average difference between each sample and its mutual nearest neighbors, and the batch-corrected data is obtained by adding this correction vector to the original values.

2.4. Mathematical Formulation of the Four Methods

To make the comparison among methods concrete, researchers summarise here the core mathematical formulation underlying each of the four correction algorithms, following the original method descriptions. ComBat. For gene g in sample j belonging to batch i , the observed expression Y_{ijg} is modeled under a location-scale framework as:

$$Y_{ijg} = \alpha_g + X\beta_g + \gamma_{ig} + \delta_{ig}\epsilon_{ijg} \quad (1)$$

where α_g is the overall expression level of gene g , X is a design matrix encoding the biological conditions of interest, β_g is the corresponding vector of regression coefficients, and ϵ_{ijg} is assumed to be normally distributed with mean zero and gene-specific variance σ^2_g . The terms γ_{ig} and δ_{ig} represent the additive and multiplicative batch effects of batch i on gene g , respectively. After estimating $\hat{\alpha}_g$, $\hat{\beta}_g$, $\hat{\gamma}_{ig}$, and $\hat{\delta}_{ig}$ from the data, the batch-adjusted value is recovered as:

$$Y^*_{ijg} = [(Y_{ijg} - \hat{\alpha}_g - X\hat{\beta}_g - \hat{\gamma}_{ig}) / \hat{\delta}_{ig}] + \hat{\alpha}_g + X\hat{\beta}_g \quad (2)$$

In practice, ComBat proceeds in five steps: (1) data preprocessing, typically log-transformation and centering; (2) fitting the location- scale model above by standard linear regression; (3) empirical Bayes estimation of the batch-effect parameters, assuming the batch-specific effects are normally distributed around zero and pooling information across all genes to stabilize these estimates; (4) shrinkage of the per-gene batch-effect estimates toward the overall mean, which reduces the influence of outlying genes and improves stability, particularly when the number of samples per batch is small; and (5) subtraction of the shrunken batch-effect estimate from each observation, performed independently per gene. Because steps 2–5 leverage information pooled across the entire dataset rather than treating each gene in isolation, ComBat tends to be comparatively robust even for moderate sample sizes, though its assumption of a single, well-behaved batch variable becomes more difficult to satisfy as the number of unmodeled confounders grows.

Surrogate Variable Analysis (SVA) begins from the general linear model:

$$Y = X\beta + ZB + \epsilon, \quad (3)$$

In this model, X contains the factors of biological interest; Z contains (potentially unobserved) variables associated with unwanted technical variation such as batch; B is the corresponding coefficient matrix; and ϵ is the residual error. Because Z is frequently unknown, SVA introduces one or more surrogate variables S estimated directly from the residuals of an initial fit of Y on X , typically via singular value decomposition, yielding the extended model:

$$Y = X\beta + ZB + S + \epsilon. \quad (4)$$

The surrogate variables are included as additional covariates in downstream analyses, allowing unmodeled sources of variation to be statistically accounted for without requiring knowledge of their physical origins in advance. SVA-seq extends this same logic to count-based RNA-Seq data by substituting negative-binomial modelling assumptions for the Gaussian assumptions underlying standard SVA, making it more appropriate for the discrete, overdispersed, and often zero-inflated nature of sequencing read counts. The number of surrogate variables to estimate is itself a tunable parameter and, in our analysis, was chosen based on the number of biological groups in each dataset.

Beclear: BEclear treats the $m \times n$ expression (or methylation) matrix D as approximately low-rank for all entries unaffected by batch effects, seeking two matrices L ($m \times r$) and R ($r \times n$) such that the product $[LR]$ closely approximates D at every unaffected entry. These matrices are estimated by gradient descent, minimising the global loss between $[LR]$ and D . Once optimisation converges, a completed matrix $D_{\text{comp}} = [LR]$ is obtained, and only those entries previously identified as batch-affected are replaced with their corresponding value in D_{comp} ; all other entries are left untouched, and any replacement value falling outside the valid data range is clipped accordingly. Identification of batch-affected genes proceeds via a weighting scheme in which each gene within a batch is assigned to a bin according to the magnitude of the difference (mdif) between its median expression in that batch and its median expression across the remaining batches, with the per-batch BE-score computed as a weighted average:

$$\text{BE-score} = (1/N) \sum_{i \in \text{mdif}} (N_{gi} \cdot w_i) \quad (5)$$

where N is the total number of genes in the batch, N_{gi} is the number of genes falling into the i -th mdif category, and w_i is the weight assigned to that category (zero for $\text{mdif} < 0.05$, increasing in increments for larger differences). This selective, entry-level replacement strategy—rather than a global rescaling of the entire matrix—preserves unaffected data exactly, which is particularly valuable for sparse, high-precision data types such as DNA methylation β -values, the original target application of BEclear.

Mutual Nearest Neighbors (MNN): Given two batches X_1 (n_1 samples) and X_2 (n_2 samples) over p shared features, MNN first identifies, for each sample in X_1 , its nearest neighbor in X_2 , and vice versa for each sample in X_2 ; a pair (i, j) is retained as a mutual nearest neighbor if sample i in X_1 is the nearest neighbor of sample j in X_2 and sample j is simultaneously the nearest neighbor of sample i , formally:

$$M = \{(i, j) : \arg\min_j |X_{1i} - X_{2j}| = i \text{ and } \arg\min_i |X_{1i} - X_{2j}| = j\} \quad (6)$$

For every sample i in batch 1 with at least one mutual nearest neighbor, a correction vector is computed as the average difference between sample i and its matched neighbors in batch 2:

$$\Delta X_{1i} = (1/|M_i|) \sum_{(i,j) \in M_i} (X_{2j} - X_{1i}) \quad (7)$$

and the batch-corrected value is obtained by adding this correction vector back to the original observation, $X'_1 = X_1 + \Delta X_1$. Because this procedure relies only on locally matched pairs of similar samples rather than on a global model of the batch effect, MNN can, in principle, accommodate batch effects whose magnitude or direction varies across different regions of expression space—a flexibility not shared by methods such as ComBat that assume a single additive/multiplicative correction per gene and batch. The practical trade-off is that fastMNN is comparatively expensive to compute for large datasets and its output (correction vectors applied within a reduced PCA subspace) is somewhat less directly interpretable in terms of the original expression units.

2.4.1. Reproducibility and Workflow Automation

To support Reproducibility across the two datasets and the several synthetic BE-injection strategies described above, researchers implemented the analysis pipeline in a Snakemake workflow. This allowed the same underlying R code—covering data loading, synthetic BE injection, and application of each of the four correction methods—to run identically across both the Tspan7 and gingival datasets without manual re-adaptation, and it ensured that intermediate outputs such as the surrogate-variable diagnostic plots could be regenerated deterministically. Beyond convenience, this workflow-based approach is intended to make the overall benchmarking strategy more readily portable to additional datasets or additional correction methods in future extensions of this work.

2.5. Evaluation

Researchers evaluated correction performance using principal component analysis (PCA) of the expression matrices before and after each correction, k-means clustering, and outlier detection. As a summary metric, researchers computed the Pearson correlation between the first two principal component loadings for the original data and each corrected dataset, with values closer to zero indicating more effective removal of batch-associated structure without introducing new correlated variation.

3. Results

3.1. Characteristics of the Original Data

A heatmap visualization of the Tspan7 dataset showed elevated expression concentrated in the final replicate column, consistent with expected experimental grouping between SCRAM controls and ShTspan7 knockdowns. The gingival dataset exhibited a more heterogeneous expression landscape, with gene AC064 showing the most distinct expression pattern across samples (Figure 2).

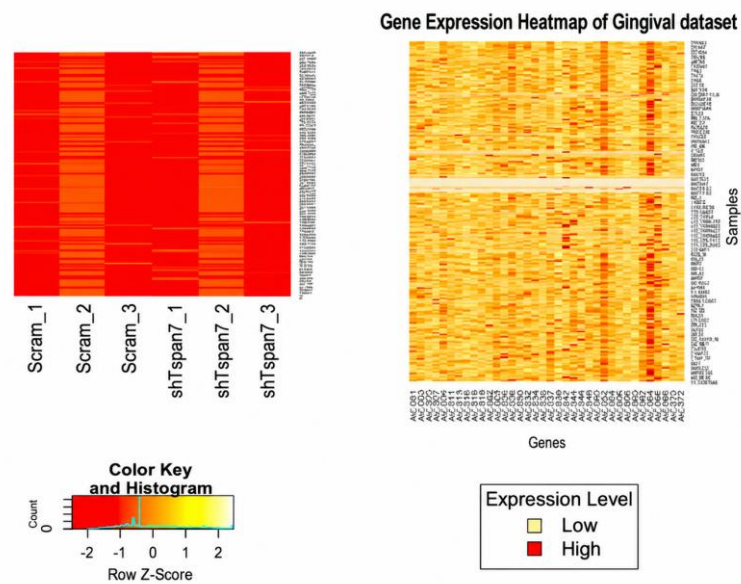


Figure 2: Gene expression heatmaps of the two original datasets before synthetic batch effect introduction. Left: Tspan7 dataset (GSE213730), showing SCRAM control and ShTspan7 knockdown replicates. Right: gingival dataset (GSE173078), showing expression across periodontal health, gingivitis, and periodontitis samples

Both datasets showed a substantial fraction of near-zero expression values alongside a smaller number of genes with very high expression (exceeding 10^5), a typical pattern for RNA-Seq count data that reflects genuine biological silencing/activation rather than technical artifact.

3.2. Synthetic Batch Effect Introduction

The additive-random-noise strategy, calibrated to the dataset's average SD (~ 8000 units) and mean (~ 839), produced favorable average read counts but tended to overinflate near-zero values while underadjusting high-count genes (Figure 3).

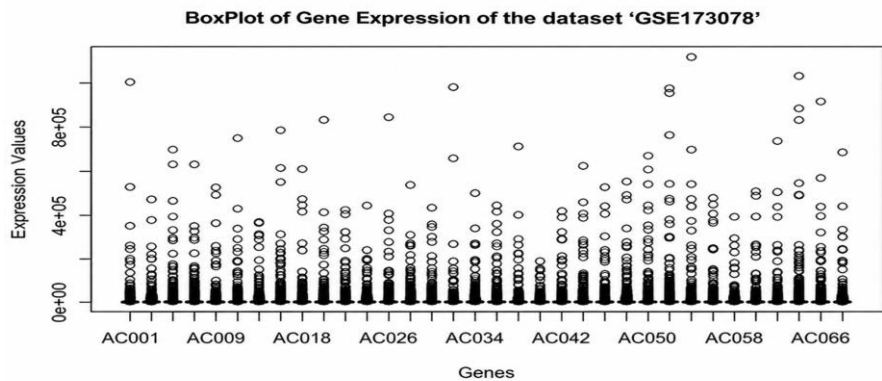


Figure 3: Distribution of gene expression values in the gingival dataset, illustrating the wide dynamic range (near-zero to $>10^5$) that the additive-noise and mean-augmentation strategies had to accommodate

The row/column mean-augmentation strategy showed a similar imbalance. The cluster-mean augmentation strategy, which identified eight clusters via the elbow method, successfully introduced structured, cluster-associated variation after excluding one singleton cluster and was judged the most representative of realistic batch structure among the three retained strategies.

3.3. Batch Effect Detection and Removal

SVA-seq identified one surrogate variable for the two-group Tspan7 dataset and four surrogate variables for the more heterogeneous gingival dataset, consistent with the respective number of underlying biological groups in each (Figure 4).

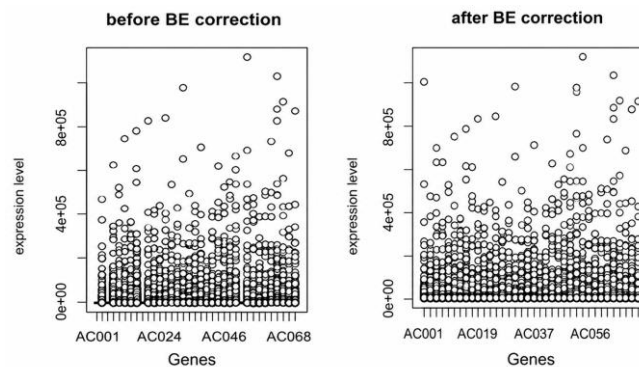


Figure 4: Gene expression values before (left) and after (right) ComBat correction, gingival dataset

ComBat flagged 3728 read counts as batch-affected in the Tspan7 dataset and 3115 in the gingival dataset (Figure 5).

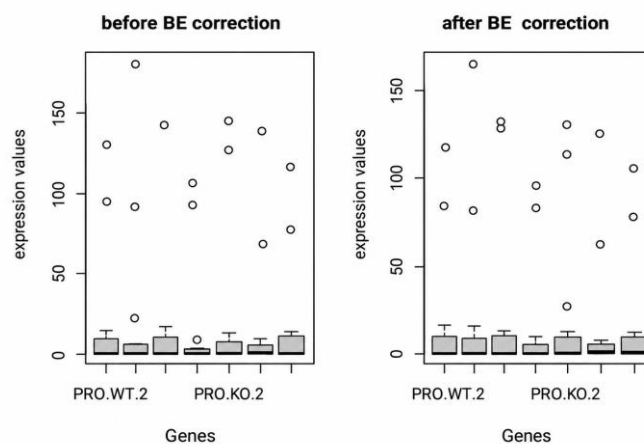


Figure 5: Expression values in a subset of the Tspan7 dataset before (left) and after (right) BEclear correction

BEclear, applied to a subset of the Tspan7 dataset, altered a comparatively small number of values (36 discrepant entries identified via the comparedf function), consistent with its design for the smaller effect sizes typical of DNA methylation data rather than RNA-Seq. (Figure 6).

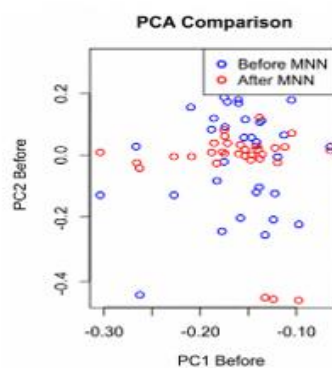


Figure 6: Principal component analysis (PC1 vs PC2) of the Tspan7 dataset before and after MNN correction

MNN produced the largest, most consistent shift in PCA space between corrected and uncorrected data, and yielded the lowest correlation between the first and second principal component loadings after correction (-1.316×10^{-5}), compared to -2.102×10^{-5} , -1.698×10^{-5} , and 1.497×10^{-5} for ComBat, SVA, and BEclear, respectively.

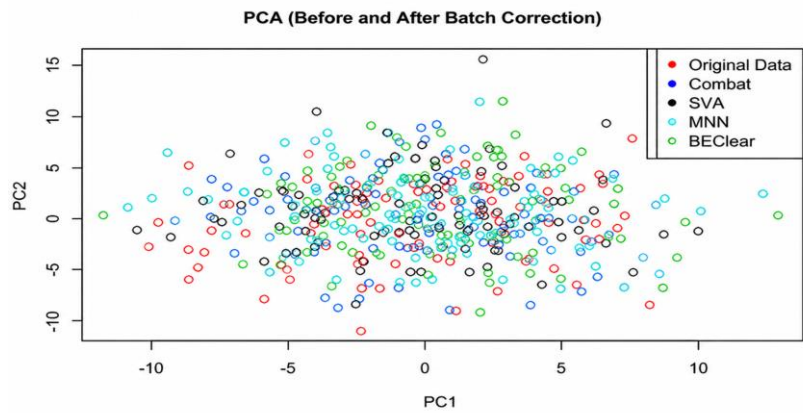


Figure 7: Principal component comparison (PC1 vs PC2) of the original data and the four batch-corrected datasets (ComBat, SVA, BEclear, MNN)

Figure 7 compares the principal components of all four corrected datasets with those of the original data, and Table 1 summarises the four methods and their relative performance in this study.

Table 1: Summary of the four batch effect removal methods applied in this study and their comparative performance

Method	Category	Approach	Relative performance in this study
ComBat	Location-scale (Empirical Bayes)	Estimates and shrinks batch-specific mean/variance parameters via empirical Bayes.	Moderate; ~3728 and 3115 read counts flagged as batch-affected in the two datasets, respectively
SVA / SVA-seq	Matrix factorisation / surrogate variables	Estimates surrogate variables capturing unmodeled variation via SVD on model residuals	Comparable to ComBat; less suited to the large-sample, high-dimensional regime used here
BEclear	Matrix factorisation (latent factor model)	Statistically detects BE-affected entries and imputes them via low-rank matrix completion.	Comparable results with limited deviation from original data (36 discrepant values); computationally demanding
MNN (fastMNN)	Distance-based (mutual nearest neighbors)	Identifies matched sample pairs across batches within a PCA subspace and adjusts expression accordingly	Best-performing; lowest PC1–PC2 correlation (-1.316×10^{-5}) after correction

4. Discussions

Across all three synthetic BE introduction strategies and both datasets, MNN and SVA emerged as the most effective correction methods, with ComBat and BEclear performing comparably but with somewhat greater residual batch-associated structure. This ordering is broadly consistent with prior comparative work: Chen et al. [8] similarly found that ComBat performs strongly overall while noting its limitations with small batch sizes, and highlighted that methods incorporating a location-shift step (such as ComBat) generally outperform those that do not. ComBat's empirical Bayes framework suits datasets with a moderate number of batches and adequate samples per batch. Still, its assumption that batch effects are attributable to a single random block effect may be too restrictive when other unmodeled sources of variation are present. SVA is more flexible in this regard, explicitly modelling unknown sources of variation as surrogate variables, but is typically more effective with smaller sample sizes than were used here. BEclear, purpose-built for DNA methylation data, showed reasonable but modest correction in this RNA-Seq context and required substantially more computation (including explicit parallelisation), consistent with its factorisation-based approach being better matched to methylation β -values than to count-based expression data. MNN's distance-based correction, which explicitly matches equivalent samples across batches before adjusting their expression,

appeared to generalize most robustly across the different synthetic BE injection strategies used here, consistent with its increasing adoption for batch integration in transcriptomic and single-cell contexts.

A more granular, method-by-method view of relative strengths and weaknesses—synthesizing our own observations with the broader comparative literature, particularly Chen et al. [8]—helps explain this ordering. Mean-centering and DWD-type approaches reliably adjust the location (mean) of each batch but do not modify the variance–covariance structure within a batch; other approaches, such as XPN and multi-batch covariance adjustment, integrate batches more explicitly and show better agreement in the eigenvalue structure. Chen et al. [8] found that ComBat, as an empirical Bayes implementation, outperformed other methods overall. Still, its performance degrades with small batch sizes, and the original algorithm handles only two batches at a time, which can complicate its application to designs with more than two experimental groups. Related ratio-based methods can perform well in specific predictive modeling contexts but risk conflating batch effects with genuine biological effects and generally underperform ComBat in accuracy and ROC-AUC. DWD can, with a stepwise extension, accommodate three or more batches, but performs poorly with small batch sizes, and its standardisation step can distort the relative scale between case and control samples, sometimes necessitating additional rescaling before results from different studies can be compared directly. SVD- and SVA-based methods are effective at identifying the presence of batch effects but can struggle to unambiguously identify which eigenvector corresponds to batch as opposed to biological variation; because SVD implicitly assumes an approximately Gaussian distribution of variation, performance can degrade when batch effects arise from heterogeneous, non-Gaussian sources, and SVA inherits this same sensitivity, being generally less robust to outliers than empirical Bayes methods.

Finally, approaches such as PAMR can achieve high measured accuracy but treat all samples uniformly, risking over- or under-correction of specific samples, and are generally less precise than ComBat in head-to-head comparisons. Several limitations should be noted. First, researchers found no existing tool purpose-built for synthetic BE injection, so our three introduction strategies, while designed to be complementary, are necessarily approximations of real-world BE structure rather than validated ground-truth generators. Second, BEclear's design assumptions (targeting sparse, methylation-like data) make it a less natural fit for dense RNA-Seq count matrices, which may partly explain its comparatively modest performance here. Third, our evaluation relied primarily on PCA-based correlation metrics and visual inspection rather than a comprehensive suite of quantitative benchmarks (e.g., silhouette scores, kBET, or downstream differential expression concordance), which would strengthen future comparative work. Finally, as Nygaard et al. [18] note, correction methods that successfully remove batch-associated variance can, in some circumstances, also exaggerate the apparent confidence of downstream analyses; consider this risk when applying any of the four methods evaluated here in a real analytical pipeline. Overall, our results support the practical recommendation that MNN and SVA are reasonable default choices for RNA-Seq batch effect correction in datasets resembling those studied here, with ComBat and BEclear remaining viable alternatives depending on batch structure, sample size, and computational constraints.

5. Conclusion

Batch effects remain one of the most persistent obstacles to reliably integrating RNA-Seq data from multiple sources, since they can obscure or mimic genuine biological signal if left unaddressed. A central limitation in this field is the absence of ground truth for real datasets. Without true, batch-free expression values, it is difficult to assess how well a given correction method performs rigorously. This study addressed that gap by developing a controlled benchmarking framework in which researchers deliberately introduced batch effects of known origin and magnitude into two independent, previously unconfounded RNA-Seq datasets, then systematically removed them using four representative correction algorithms. Researchers designed and implemented three complementary strategies for synthetic batch effect injection—additive random noise scaled to the data's standard deviation, row/column mean augmentation, and cluster-mean augmentation guided by k-means partitioning—since no existing tool served this purpose directly. Each strategy introduced a structurally distinct form of technical variation, allowing the four correction methods (ComBat, SVA, BEclear, and MNN) to be evaluated across a broader range of batch-effect scenarios than a single injection method would have permitted. All four correction methods measurably reduced batch-associated variance relative to uncorrected data, confirming that each can detect and mitigate synthetic batch structure in RNA-Seq count data.

Among them, MNN consistently achieved the strongest and most reproducible correction, as reflected in the lowest post-correction correlation between principal components, and is therefore the most broadly effective option for the datasets and batch-effect types examined here. SVA performed comparably well and offers a more flexible, assumption-light alternative when the source of technical variation is not fully known in advance. ComBat and BEclear, while somewhat less effective in this high-dimensional RNA-Seq setting, remain reliable and computationally efficient choices, particularly for data types that more closely match their original design targets—moderate batch sizes with adequate replication for ComBat, and DNA methylation-like data for BEclear. Taken together, these findings reinforce the broader conclusion that no single batch-correction method is universally optimal: the best choice depends on the number and size of batches, the data's dimensionality,

and how well the source of technical variation is understood. The synthetic batch-effect injection framework introduced here offers a practical, reusable strategy for benchmarking correction methods under controlled, known conditions. It could be readily extended to additional datasets, effect types, and correction algorithms in future work. As a next step, incorporating this benchmarking approach as a batch-effect-aware preprocessing module within downstream analysis pipelines, such as PPIXpress, would help ensure that integrative RNA-Seq analyses are not confounded by uncorrected or overcorrected technical variation before biological interpretation begins.

Acknowledgment: The authors gratefully acknowledge Saarland University, Chia Nan University of Pharmacy and Science, S Khan Lab, and Filial of Medical Sciences of Holguín for their valuable support, guidance, research resources, and institutional assistance throughout this research.

Data Availability Statement: The data supporting this study are available from the corresponding author and may be provided upon reasonable request.

Funding Statement: No external funding or financial assistance was received for conducting this research or preparing this manuscript.

Conflicts of Interest Statement: The authors declare that there are no conflicts of interest related to this study. This manuscript represents an original contribution, and all sources of information used have been properly acknowledged through appropriate citations and references.

Ethics and Consent Statement: The study was conducted in accordance with ethical guidelines. Participants were assured that their responses would be kept confidential and anonymous.

References

1. M. F. Adamer, S. C. Brüningk, A. Tejada-Arranz, F. Estermann, M. Basler, and K. M. Borgwardt, “reComBat: Batch effect removal in large-scale, multi-source omics data integration,” *bioRxiv*, 2021. [Accessed by 07/07/2024].
2. R. Akulenko, M. Merl, and V. Helms, “BEclear: Batch effect detection and adjustment in DNA methylation data,” *PLoS ONE*, vol. 11, no. 8, pp. 1–17, 2016.
3. O. Alter, P. O. Brown, and D. Botstein, “Singular value decomposition for genome-wide expression data processing and modeling,” *Proceedings of the National Academy of Sciences*, vol. 97, no. 18, pp. 10101–10106, 2000.
4. F. Bayat, “Variance-stabilized signals for biological data,” *Simon Fraser University*, 2022. [Accessed by 09/07/2024].
5. F. Bayat and M. Libbrecht, “VSS: variance-stabilized signals for sequencing-based genomic signals,” *Bioinformatics*, vol. 37, no. 23, pp. 4383–4391, 2021.
6. M. Benito, J. Parker, Q. Du, J. Wu, D. Xiang, C. M. Perou, and J. S. Marron, “Adjustment of systematic microarray data biases,” *Bioinformatics*, vol. 20, no. 1, pp. 105–114, 2004.
7. C. Lazar, S. Meganck, J. Taminiau, D. Steinhoff, A. Coletta, C. Molter, D. Y. Weiss-Solís, R. Duque, H. Bersini, and A. Nowé, “Batch effect removal methods for microarray gene expression data integration: A survey,” *Briefings in Bioinformatics*, vol. 14, no. 4, pp. 469–490, 2013.
8. C. Chen, K. Grennan, J. Badner, D. Zhang, E. Gershon, L. Jin, and C. Liu, “Removing batch effects in analysis of expression microarray data: An evaluation of six batch adjustment methods,” *PLoS ONE*, vol. 6, no. 2, pp. 1–10, 2011.
9. Y. Zhang, G. Parmigiani, and W. E. Johnson, “ComBat-seq: Batch effect adjustment for RNA-seq count data,” *NAR Genomics and Bioinformatics*, vol. 2, no. 3, pp. 1–10, 2020.
10. W. E. Johnson, C. Li, and A. Rabinovic, “Adjusting batch effects in microarray expression data using empirical Bayes methods,” *Biostatistics*, vol. 8, no. 1, pp. 118–127, 2007.
11. J. A. Lee, K. K. Dobbin, and J. Ahn, “Covariance adjustment for batch effect in gene expression data,” *Statistics in Medicine*, vol. 33, no. 15, pp. 2681–2695, 2014.
12. J. T. Leek and J. D. Storey, “Capturing heterogeneity in gene expression studies by surrogate variable analysis,” *PLoS Genetics*, vol. 3, no. 9, pp. 1–12, 2007.
13. J. A. Gagnon-Bartsch and T. P. Speed, “Using control genes to correct for unwanted variation in microarray data,” *Biostatistics*, vol. 13, no. 3, pp. 539–552, 2012.
14. J. T. Leek, W. E. Johnson, H. S. Parker, A. E. Jaffe, and J. D. Storey, “The sva package for removing batch effects and other unwanted variation in high-throughput experiments,” *Bioinformatics*, vol. 28, no. 6, pp. 882–883, 2012.
15. J. Luo, M. Schumacher, A. Scherer, D. Sanoudou, D. Megherbi, T. Davison, T. Shi, W. Tong, L. Shi, H. Hong, C. Zhao, F. Elloumi, W. Shi, R. Thomas, S. Lin, G. Tillinghast, G. Liu, Y. Zhou, D. Herman, Y. Li, Y. Deng, H. Fang, P. Bushel, M. Woods, and J. Zhang, “A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data,” *The Pharmacogenomics Journal*, vol. 10, no. 7, pp.

- 278–291, 2010.
16. J. S. Marron, “Distance-weighted discrimination,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 2, pp. 109–114, 2015.
 17. C. Müller, A. Schillert, C. Röthmeier, D. A. Trégouët, C. Proust, H. Binder, N. Pfeiffer, M. Beutel, K. J. Lackner, R. B. Schnabel, L. Tiret, P. S. Wild, S. Blankenberg, T. Zeller, and A. Ziegler, “Removing batch effects from longitudinal gene expression—quantile normalization plus ComBat as best approach for microarray transcriptome data,” *PLoS ONE*, vol. 11, no. 6, pp. 1–23, 2015.
 18. V. Nygaard, E. A. Rødland, and E. Hovig, “Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses,” *Biostatistics*, vol. 17, no. 1, pp. 29–39, 2016.
 19. B. Phipson, J. Maksimovic, and A. Oshlack, “missMethyl: An R package for analyzing data from Illumina’s HumanMethylation450 platform,” *Bioinformatics*, vol. 32, no. 2, pp. 286–288, 2016.
 20. D. Risso, K. Schwartz, G. Sherlock, and S. Dudoit, “GC-content normalization for RNA-Seq data,” *BMC Bioinformatics*, vol. 12, no. 12, pp. 1–17, 2011.
 21. M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth, “limma powers differential expression analyses for RNA-sequencing and microarray studies,” *Nucleic Acids Research*, vol. 43, no. 7, pp. 1–13, 2015.
 22. J. Rudy and F. Valafar, “Empirical comparison of cross-platform normalization methods for gene expression data,” *BMC Bioinformatics*, vol. 12, no. 12, pp. 1–22, 2011.
 23. A. Scherer, “Batch effects and noise in microarray experiments: Sources and Solutions,” *WILEY*, Hoboken, New Jersey, United States of America, 2009.
 24. A. A. Shabalín, H. Tjelmeland, C. Fan, C. M. Perou, and A. B. Nobel, “Merging two gene-expression studies via cross-platform normalization,” *Bioinformatics*, vol. 24, no. 9, pp. 1154–1160, 2008.
 25. A. H. Sims, G. J. Smethurst, Y. Hey, M. J. Okoniewski, S. D. Pepper, A. Howell, C. J. Miller, and R. B. Clarke, “The removal of multiplicative, systematic bias allows integration of breast cancer gene expression datasets—improving meta-analysis and prediction of prognosis,” *BMC Medical Genomics*, vol. 1, no. 9, pp. 1–14, 2008.
 26. J. Taminau, “Unlocking the potential of public available gene expression data for large-scale analysis,” *Brussels University Press*, Brussels, Belgium, 2012.
 27. H. T. N. Tran, K. S. Ang, M. Chevrier, X. Zhang, N. Y. S. Lee, M. Goh, and J. Chen, “A benchmark of batch-effect correction methods for single-cell RNA sequencing data,” *Genome Biology*, vol. 21, no. 1, pp. 1–32, 2020.
 28. M. Ugidos, M. J. Nueda, J. M. Prats-Montalbán, A. Ferrer, A. Conesa, and S. Tarazona, “MultiBaC: an R package to remove batch effects in multi-omic experiments,” *Bioinformatics*, vol. 38, no. 9, pp. 2657–2658, 2022.
 29. H. Ye, X. Zhang, C. Wang, E. L. Goode, and J. Chen, “Batch-effect correction with sample remeasurement in highly confounded case-control studies,” *Nature Computational Science*, vol. 3, no. 8, pp. 709–719, 2023.