bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

# BatchServer: a web server for batch effect evaluation, visualization and correction

Tiansheng Zhu[1,2,3], Guo-Bo Chen[4], Chunhui Yuan[2,3], Rui Sun[2,3], Fangfei Zhang[2,3], Xiao Yi[2,3], Shuigen Zhou[1,*], Tiannan Guo[2,3,*]

[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, China. [2]Key Laboratory of Structural Biology of Zhejiang Province, School of Life Sciences, Westlake University, 18 Shilongshan Road, Hangzhou 310024, Zhejiang Province, China. [3]Institute of Basic Medical Sciences, Westlake Institute for Advanced Study, 18 Shilongshan Road, Hangzhou 310024, Zhejiang Province, China. [4]Clinical Research Institute, Zhejiang Provincial People's Hospital, People's Hospital of Hangzhou Medical College, Hangzhou, Zhejiang, China

*To whom correspondence should be addressed.

Shuigen Zhou: sgzhou@fudan.edu.cn;

Tiannan Guo: guotiannan@westlake.edu.cn

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

## Abstract

**Background**: Batch effects are unwanted data variations that may obscure biological signals, leading to bias or errors in subsequent data analyses. Effective evaluation and elimination of batch effects is thus necessary for omics data analysis, especially in the context of large cohort of thousands of samples with different experimental platforms. Existing batch effect reducing tools mainly focus on the development of algorithms, while requiring programming skills and the knowledge of data distribution limits their application for many researchers. In order to facilitate evaluation and correction of batch effects, we provided an user-friendly and easy-to-use graphical batch effects analysis web platform.

**Results**: We developed an open-source R/Shiny based web server -- BatchServer that allows users to graphical interactively evaluate, visualize and correct of the batch effects in high-throughput data sets. BatchServer including a modified ComBat, which was a popular batch effect adjustment tool to correct batch effects, PVCA (Principal Variance Component Analysis) and UMAP (Manifold Approximation and Projection) to evaluate and visualize batch effects. BatchServer is an efficient batch effects processing platform, as its application in three publicly available data sets.

**Conclusion**: Our user-friendly online open-source web server BatchServer supports comprehensive batch effects analysis facilitating the batch effect evaluations and corrections for biologists. BatchServer is deployed at https://lifeinfo.shinyapps.io/batchserver/ as a web server. The source codes are freely available at https://github.com/zhutiansheng/batch_server.

**Keywords:** Batch effects, Data pre-processing, ComBat, PVCA, UMAP

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

## Background

High-throughput omics data are usually generated with unwanted systematic variation or so-called "batch effects". Batch effects are unwanted sources of systematic variation confounded with biological signals when samples are processed and measured in different batches due to different laboratory conditions, reagent lots and personnels [1, 2]. If not handled properly, batch effects may distort biological signals [3] and mislead downstream analyses with high false positive rate [4]. Evaluation and correction of batch effects is essential in analyzing large-scale omics data.

Assessing batch effects precedes omics-data analysis. Principal variance component analysis (PVCA) [5] and uniform manifold approximation and projection (UMAP)[6] are two widely adopted methods to identify or visualize batch effects. PVCA leverages principal component analysis (PCA) [7] and variance components analysis (VCA), and fits a mixed linear model to estimate the proportion of variation of each factor. PVCA has been used to evaluate the effectiveness of batch effect correction [5]. UMAP is an emerging non-linear dimensionality reduction method, reported as the state-of-art tool for visualization of single cell cytometry and transcriptome data in terms of run times, reproducibility and organization of single cell clusters [6]. We applied these two methods to identify quantify or batch effect.

The most intuitive way to correct batch effects is normalization technique, which adjusts the global properties of the data by comparing individual samples. However, batch effects may affect features in different extents, so simply exploiting normalization is sometimes insufficient to remove complex batch effects induced by multiple features [1]. To effectively eliminate batch effects, several methods have been developed, including singular value decomposition (SVD) [8], surrogate variable analysis (SVA) [9], exploBATCH [10], BatchI [11] and ComBat [12]. Among those methods, ComBat, based on a parametric or non-parametric empirical Bayes strategy, is arguably the most widely method for batch correction. ComBat has been reported to be reliable and robust to outliers when tested over a

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

large number of samples [2, 13]. However, User needs to determine whether to use parametric based method upon the distribution of data that is usually unknown [12]. Here, we automated this switch for ComBat with goodness of fit test. We further developed an open source web server – BatchServer that integrates PVCA, UMAP and autoComBat to provide researchers with easy-to-use interface to evaluate and correct potential batch effects in large-scale omics data.

## Implementation

### Architecture

The architecture of BatchServer consists of three layers (**Figure 1**): 1) Data input layer is provided for uploading input files of data files and sample information files and for interactively selecting batches and co-variate names; 2) Data processing is responsible for batch effect estimation, visualization and correction; 3) Data output layer can, used to display and download data processing results. BatchServer is written in R and calls several R packages, including Shiny and Shinydashboard for R/Shiny web interface; pvca for batch effect evaluation; plotly, umap and ggplot2 for batch effect visualization; sva for batch effect correction. BatchServer is inherited from the interactive microservice framework Shiny, integrating web interface in app.R, ui.R, and server.R which are called through global.R. BatchServer provided an easy-to-use interactive user interface.

### Imputation of missing values

Missing values are common in omics data sets due to technical or biological issues [14]. The mechanisms for missing values occurrence are complex and can be estimated in a number of ways [15]. Most statistical and machine learning methods do not allow large portion of missing values; therefore, BatchServer provides four computationally efficient ways to replace missing values by ‘1’, ‘0’, ‘10% of minimum’, or ‘minimum’, where minimum is the minimal value in the upload data matrix.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

## Automated ComBat

ComBat is essentially a linear model that attempts to eliminate batch effects directly from the data sets. $Y_{ijg}$ is defined to represent the feature (e.g. protein or gene) g of sample j from batch i. Define a model that assumes

$$Y_{ijg} = \alpha_g + X\beta_g + \gamma_{ig} + \delta_{ig}\epsilon_{ijg}, \tag{1}$$

here $\alpha_g$ is the overall value of g, X is a design matrix for sample conditions, and $\beta_g$ is the vector of the regression coefficients. The error term, $\epsilon_{ijg}$, is assumed to follow a normal distribution $N{\sim}(0, \sigma_g^2)$. $\gamma_{ig}$ and $\delta_{ig}$ represent the additive and multiplicative batch effects of batch i for feature g, respectively. If (2) holds, ComBat will use the parametric Bayes method, otherwise it will use the non-parametric Bayes method to estimate $\gamma_{ig}$ and $\delta_{ig}^2$.

$$\gamma_{ig} \sim \text{N}\ (\gamma_i,\ \tau_i^2), \text{ and } \delta_{ig}^2 \sim \text{Inverse Gamma}\ (\lambda_i,\ \theta_i) \tag{2}$$

The hyperparameters $\gamma_i$, $\tau_i^2$, $\lambda_i$, $\theta_i$ are estimated empirically from data using the method of moments[12]. When using ComBat, users must manually modify the default parameters to determine whether to use the parametric or non-parametric empirical Bayes method. For ease of use, we employed Kolmogorov-Smirnov Goodness of Fit Test (K-S) to test whether the additive parameter $\gamma_{ig}$ fits normal distribution and the multiplicative parameter $\delta^2{}_{ig}$ fits inverse gamma distribution for each batch at the significance level of 0.05 [16], respectively. In other words, BatchServer automatically switches between parametric or non-parametric Bayes method upon the detected distribution underlying the input data.

Compared with the original ComBat, the improved ComBat is integrated into BatchServer with only one parameter (i.e. “par.prior”) different. The improved ComBat added an option ‘auto’ for this parameter, which automatically determines the use of parametric or non-parametric Bayes method by the above described strategy. For a better visual experience, BatchServer not only plots all batches

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

interactively, but also improves the visualization by highlighting and changing the colors of lines and points. Using the default parameters, we can see the prior plot of $\hat{\gamma}$ and $\hat{\delta}$ of batch ‘1’ which passed the K-S test (red and blue lines, **Figure 2A**). Batch ‘4’ did not pass the K-S test indicating this batch is not good (**Figure 2B**).

## Application of PVCA and UMAP

As a dimension reduction technique, UMAP was integrated into BatchServer to visualize and judge the batch effects of high dimensional omics data. The PVCA method assesses the proportion of each source of variability in a given data set. Nevertheless, PVCA in R Bioconductor only accepts ExpressionSet objects, which greatly limits its applicability other types of data object. We thus wrapped the pvcaBatchAcessess function in the pvca package and simplified the input data format. To enhance its usability, we integrated it into our web server.

## Results

### BatchServer usage

The detailed instructions for BatchServer are provided in the Readme section of the online web page. We also provide the flow diagram of using BatchServer in Figure 3. For data input, a data file and a sample information file are required. The format of these files can be tab-delimited, space-separated, comma-delimited or an Excel file. BatchServer provides test data files provided by the bladderbatch package in the web readme page. The user can upload these two files in the “Data Input” menu, then click the “Submit” button. The data read module will read, process, and store the files for subsequent uses. It is recommended for users to evaluate whether the data have batch effects using PVCA or UMAP with the online server. Both methods will display the visualization of batch effects. Once the

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

batch effect is present, it can be adjusted using improved ComBat. Users can check and download the results of batch effect evaluation. The corrected data can be also downloaded.

## Real data application

Three data sets were employed to test the performance of BatchServer. Each data set was analyzed and evaluated for time consumption using the improved ComBat. The parameters were set as par.prior = TRUE for parametric estimation, par.prior = FALSE for non-parametric and par.prior = auto for automatic detection of parametric or non-parametric estimation. Other parameters were set by default. The data before and after ComBat adjusted were analyzed and plotted by pvca and umap module.

The first set of data are microarray-based transcriptomic data (GSE19804 and GSE10072) of 227 lung cancer disease (118 cancers and 109 controls) in female nonsmokers [2]. We combined these two data matrices on their shared probes, naturally bringing out a data that were generated from two distinct batches (we named bath1 and bath2). Figure 4A showed an estimation of proportions of each factor by pvca: 1) before using ComBat (no correction) 71.57% variation was attributed to batch effect, 16.87% to biological signals (tumor and normal), 0.73% to interaction between batch and biological types, and10.84% left to residual variation; 2) after ComBat correction, batch effect variation reduced to nearly zero and consequently intensified biological variation considerably. Figure 4B and C showed consisted effect with A when using UMAP. The results were the same when using nonparametric and auto parameters (Figure 4A, B, C) because neither batch passed the K-S test (Figure 4D). Although the computational cost for method choice was negligible for combat (Additional file 2, Supplementary Table 1), the nonparametric estimation was always computational expensive than the parametric one. We also tested another two datasets. Both showed substantial batch effects as visual-

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

ized by our batchServer, but after applying automated procedures the batch effects were well controlled, leading to enhanced biological signals. Details were presented in additional file 2.

## Discussion

Although parametric and nonparametric ComBat often brought out similar results, the parametric Bayes implantation was much computationally faster than its nonparametric one. As proposed in this study, the choice of using parametric or non-parametric Bayes implementation should be upon the goodness-of-fit test for the underlying distribution of the data set. Secondly, to extend batch effect method, we used feature selection strategies to filter out batch-specific features, inspired by the method of PCA and SVD that deal with batch effects by filtering out eigen features (e.g. protein, gene, and metabolite) that are inferred as batch affect factors[12]. Finally, we developed an online web server named BatchServer, especially dedicated to evaluation, visualization, and correction of batch effects, including quantifying the proportion of variation of batch effects using PVCA, visualizing batch effects using UMAP, and correcting batch effects using the improved ComBat with goodness-of-fit test.

## Conclusions

We developed a web server called BatchServer to facilitate the evaluation, visualization, and correction of batch processing results for large-scale omics data sets. The automated ComBat can automatically select parametric or non-parametric empirical Bayes methods for batch calibration. It also integrates PVCA and UMAP to evaluate and visualize potential batch effects. BatchServer has an R/Shiny graphical user interface for enhanced usability and is easy to install on a personal computer or server, thus providing a convenient service to process batch effect for the community.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

## Availability and requirements

Project name: BatchServer (Batch effects server)

Project home page: https://lifeinfo.shinyapps.io/batchserver/

Operating system: Linux, Mac OS, and Windows.

Programming language: R

Other requirements: Tested with Chrome, Firefox and IE browsers

License: Freely available to academic researchers.

Any restrictions to use by non-academics: None

## Availability of data and materials

The proposed BatchServer and its manual are deployed at https://lifeinfo.shinyapps.io/batchserver/.

Source codes and testing data are freely available at https://github.com/zhutiansheng/batch_server

## List of abbreviations

PCA: principal components analysis; PVCA: Principal Variance Component Analysis; UMAP: Manifold Approximation and Projection; SVD: Singular Value Decomposition; t-SNE: t-distributed stochastic neighbor embedding; K-S test: Kolmogorov–Smirnov test.

## Declarations

Ethics approval and consent to participate

Not applicable.

## Competing interests

The authors declare that they have no competing interests.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

## Consent for publication

Not applicable.

## Funding

This work was supported by National Natural Science Foundation of China (General Program) (Grant No. 81972492 to T.G.), Zhejiang Provincial Natural Science Foundation for Distinguished Young Scholars (Grant No. LR19C050001 to T.G.), Hangzhou Agriculture and Society Advancement Program (Grant No. 20190101A04 to T.G.).

## Authors' contributions

TZ implemented the web application; TG, SZ, and TZ wrote the manuscript and conceived the project; GC, CY and FZ revised and modified the manuscript and provided many useful suggestions. RS and XY conducted the software testing. All authors have read and approved the final manuscript.

## Acknowledgements

Not applicable.

## References

1. Leek JT, Scharpf RB, Bravo HC, Simcha D, Langmead B, Johnson WE, Geman D, Baggerly K, Irizarry RA: **Tackling the widespread and critical impact of batch effects in high-throughput data**. *Nat Rev Genet* 2010, **11**(10):733-739.
2. Lazar C, Meganck S, Taminau J, Steenhoff D, Coletta A, Molter C, Weiss-Solis DY, Duque R, Bersini H, Nowe A: **Batch effect removal methods for microarray gene expression data integration: a survey**. *Brief Bioinform* 2013, **14**(4):469-490.
3. Goh WWB, Wong L: **Advanced bioinformatics methods for practical applications in proteomics**. *Brief Bioinform* 2019, **20**(1):347-355.
4. Chen C, Grennan K, Badner J, Zhang D, Gershon E, Jin L, Liu C: **Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods**. *PLoS One* 2011, **6**(2):e17238.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

5. Boedigheimer MJ, Wolfinger RD, Bass MB, Bushel PR, Chou JW, Cooper M, Corton JC, Fostel J, Hester S, Lee JS *et al*: **Sources of variation in baseline gene expression levels from toxicogenomics study control animals across multiple laboratories**. *Bmc Genomics* 2008, **9**:285.
6. Becht E, McInnes L, Healy J, Dutertre CA, Kwok IWH, Ng LG, Ginhoux F, Newell EW: **Dimensionality reduction for visualizing single-cell data using UMAP**. *Nat Biotechnol* 2018, **37**(1):38-44.
7. Wold S: **Principal component analysis**. 1987, **2**(1):37-52.
8. Alter O, Brown PO, Botstein D: **Singular value decomposition for genome-wide expression data processing and modeling**. *Proc Natl Acad Sci U S A* 2000, **97**(18):10101-10106.
9. Leek JT, Storey JD: **Capturing heterogeneity in gene expression studies by surrogate variable analysis**. *PLoS Genet* 2007, **3**(9):1724-1735.
10. Nyamundanda G, Poudel P, Patil Y, Sadanandam A: **A Novel Statistical Method to Diagnose, Quantify and Correct Batch Effects in Genomic Studies**. *Sci Rep* 2017, **7**(1):10849.
11. Papiez A, Marczyk M, Polanska J, Polanski A: **BatchI: Batch effect Identification in high-throughput screening data using a dynamic programming algorithm**. *Bioinformatics* 2019, **35**(11):1885-1892.
12. Johnson WE, Li C, Rabinovic A: **Adjusting batch effects in microarray expression data using empirical Bayes methods**. *Biostatistics* 2007, **8**(1):118-127.
13. Goh WWB, Wang W, Wong L: **Why Batch Effects Matter in Omics Data, and How to Avoid Them**. *Trends Biotechnol* 2017, **35**(6):498-507.
14. McGurk KA, Dagliati A, Chiasserini D, Lee D, Plant D, Baricevic-Jones I, Kelsall J, Eineman R, Reed R, Geary B *et al*: **The use of missing values in proteomic data-independent acquisition mass spectrometry to enable disease activity discrimination**. *Bioinformatics* 2019.
15. Wei R, Wang J, Su M, Jia E, Chen S, Chen T, Ni Y: **Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data**. *Sci Rep* 2018, **8**(1):663.
16. MA S: **Tests Based on EDF Statistics in Goodness-of-Fit Techniques**. *Marcel Dekker* 1986:97-194.

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

## Figure legends

Figure 1. The architecture of BatchServer. Users can submit data and sample information files via the data input layer. Once submitted, files are processed by the data process layer. Thereafter, the user is required to specify potential batch effect columns and interactively set variables, if any. The data processing layer then calculates and returns the results. Finally, the data output layer will display the results interactively through  and tables.

Figure 2. The screenshot of prior plot of γ □ and δ □ of (A) batch '1' which passed the K-S test and (B) batch '4' which didn't pass the K-S test using BatchServer. The first 50 rows of bladder data in bladderbatch Bioconductor package were used as input data.

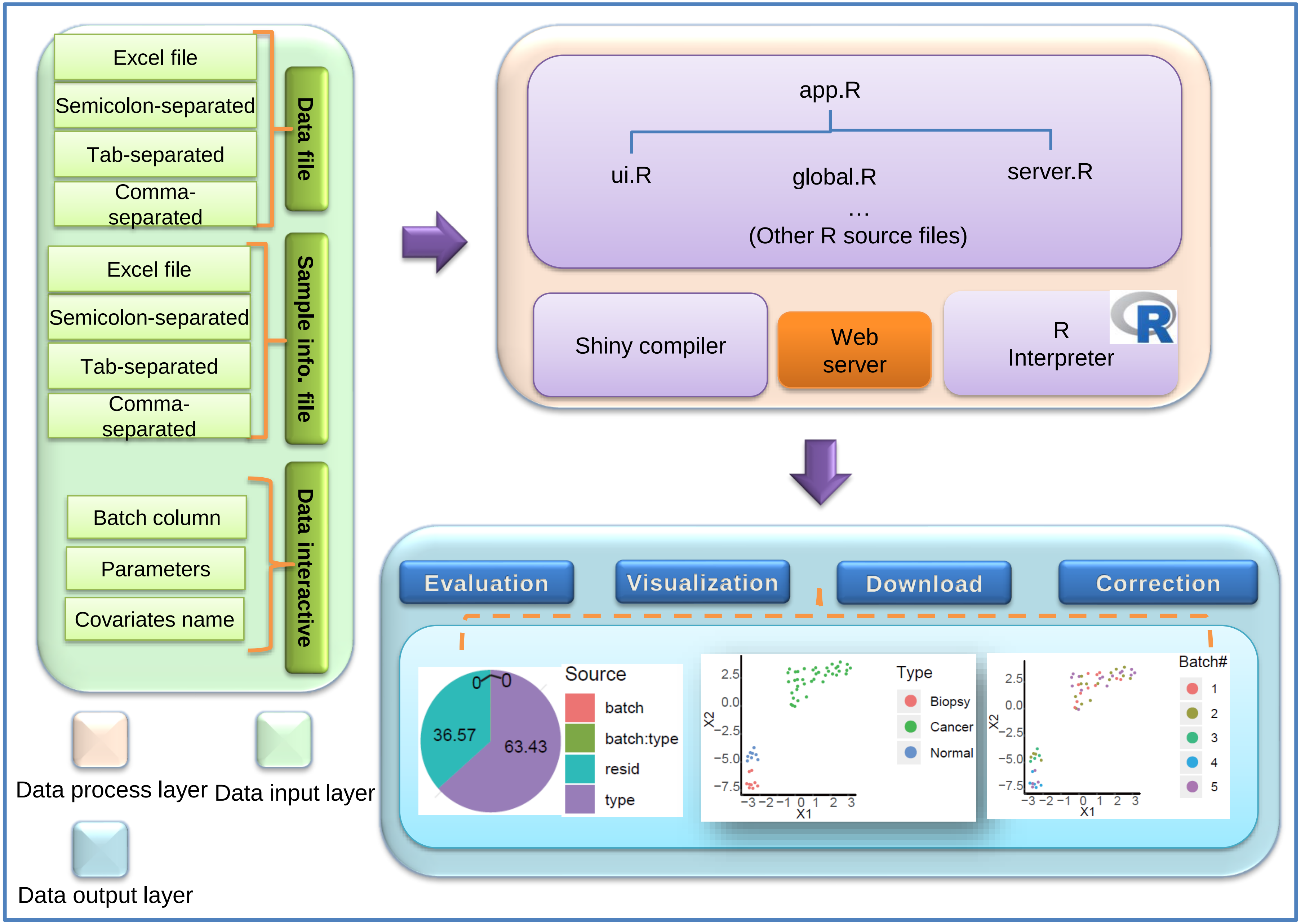
Figure 3. Flow diagram of BatchServer.

Figure 4. Improved performance of ComBat by the first data set. (A) Pie plots of batch effect using PVCA with no correction and par.prior set to nonparametric, parametric or auto for improved Com-Bat. (BC) UMAP plots show the clustering of biological and batch effect, respectively. D) Priori plots of batch effect by improved ComBat.

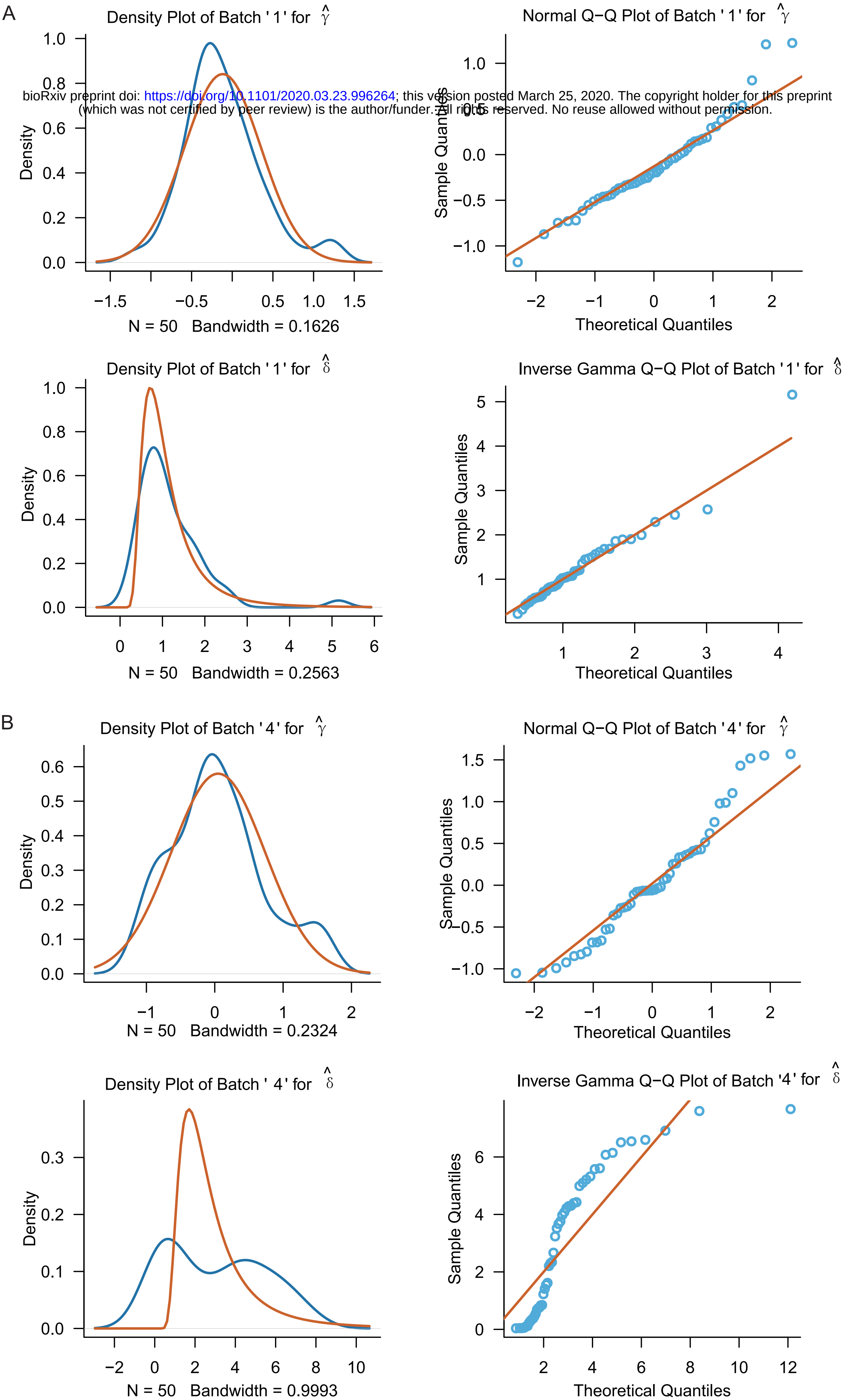
## Additional Files

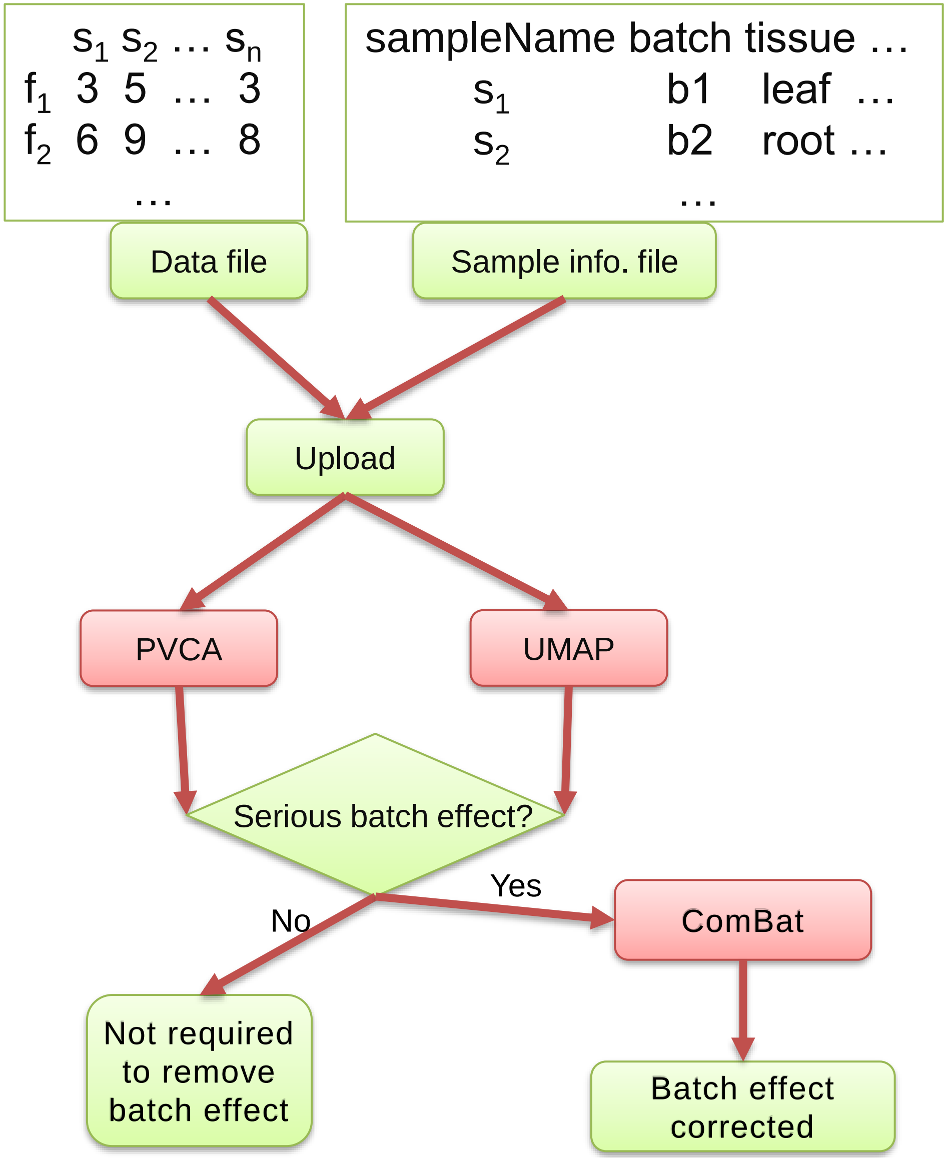
Additional file 1. A zip archive containing the source codes of BatchServer.

Additional file 2. A zip archive containing the source codes of BatchServer.


Excel file
Semicolon-separated
Tab-separated
Comma-separated
Data file
Excel file
Semicolon-separated
Tab-separated
Comma-separated
Sample info. file
Batch column
Parameters
Covariates name
Data interactive
app.R
ui.R
global.R
server.R
…
(Other R source files)
Shiny compiler
Web server
R Interpreter
Evaluation
Visualization
Download
Correction
Source
batch
batch:type
resid
type
36.57
63.43
0
0
Type
Biopsy
Cancer
Normal
Batch#
1
2
3
4
5
X1
X2
Data process layer
Data input layer
Data output layer

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.





|  | $s_1$ | $s_2$ | … | $s_n$ |
|---|---|---|---|---|
| $f_1$ | 3 | 5 | … | 3 |
| $f_2$ | 6 | 9 | … | 8 |
|  |  | … |  |  |

| sampleName | batch | tissue | … |
|---|---|---|---|
| $s_1$ | b1 | leaf | … |
| $s_2$ | b2 | root | … |
|  | … |  |  |

Data file

Sample info. file

Upload

PVCA

UMAP

Serious batch effect?

No

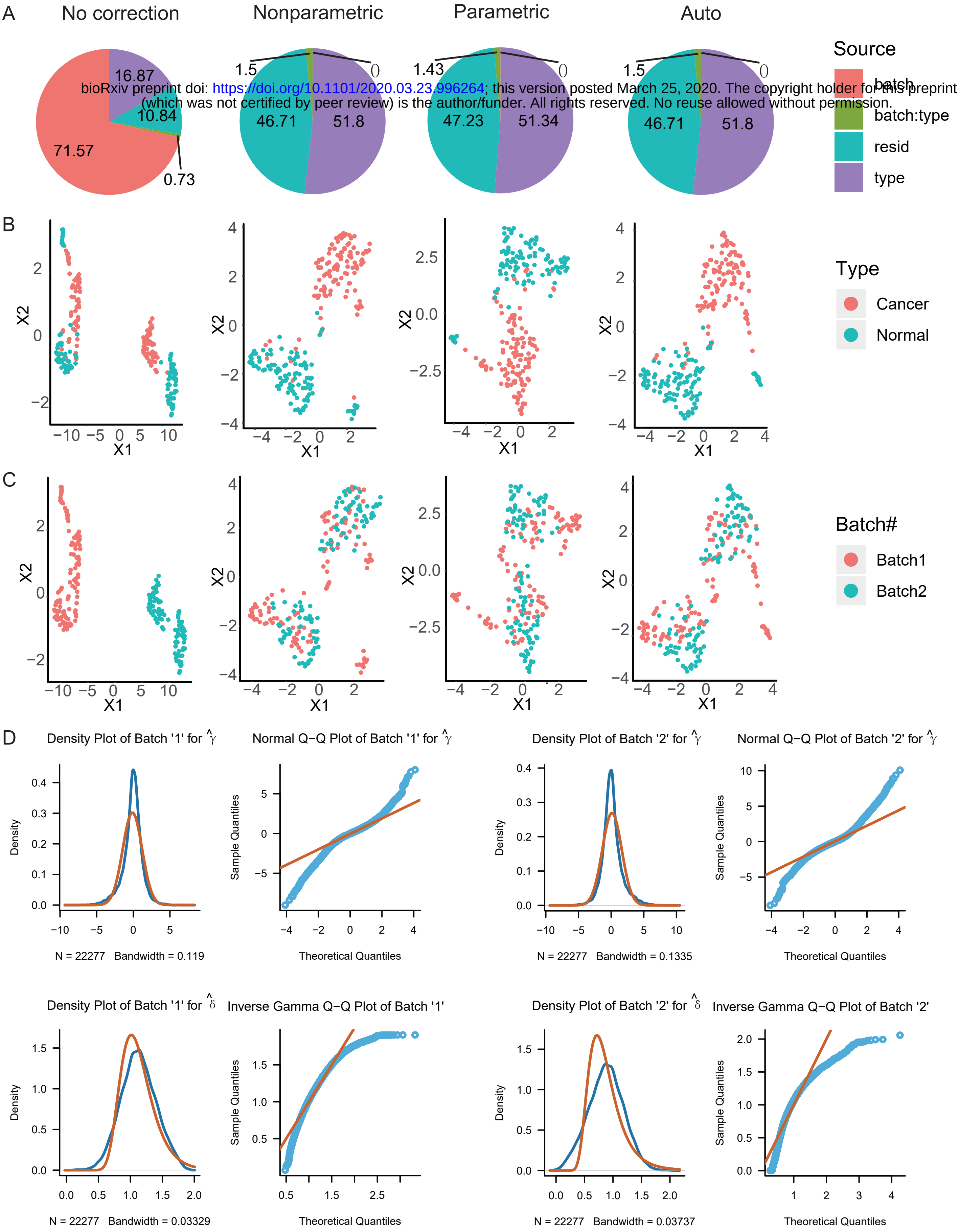
Yes

ComBat

Not required to remove batch effect

Batch effect corrected

bioRxiv preprint doi: https://doi.org/10.1101/2020.03.23.996264; this version posted March 25, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.


A
No correction
Nonparametric
Parametric
Auto
Source
batch
batch:type
resid
type
B
Type
Cancer
Normal
C
Batch#
Batch1
Batch2
X1
X2
D
Density Plot of Batch '1' for $\hat{\gamma}$
Normal Q−Q Plot of Batch '1' for $\hat{\gamma}$
Density Plot of Batch '2' for $\hat{\gamma}$
Normal Q−Q Plot of Batch '2' for $\hat{\gamma}$
N = 22277 Bandwidth = 0.119
N = 22277 Bandwidth = 0.1335
Density Plot of Batch '1' for $\hat{\delta}$
Inverse Gamma Q−Q Plot of Batch '1'
Density Plot of Batch '2' for $\hat{\delta}$
Inverse Gamma Q−Q Plot of Batch '2'
N = 22277 Bandwidth = 0.03329
N = 22277 Bandwidth = 0.03737
Density
Sample Quantiles
Theoretical Quantiles