scientific reports

Check for updates

OPEN

Comparison of the effectiveness of different normalization methods for metagenomic cross-study phenotype prediction under heterogeneity

Beibei Wang[1,2,3], Fengzhu Sun[4] & Yihui Luan[1,2,3]✉

The human microbiome, comprising microorganisms residing within and on the human body, plays a crucial role in various physiological processes and has been linked to numerous diseases. To analyze microbiome data, it is essential to account for inherent heterogeneity and variability across samples. Normalization methods have been proposed to mitigate these variations and enhance comparability. However, the performance of these methods in predicting binary phenotypes remains understudied. This study systematically evaluates different normalization methods in microbiome data analysis and their impact on disease prediction. Our findings highlight the strengths and limitations of scaling, compositional data analysis, transformation, and batch correction methods. Scaling methods like TMM show consistent performance, while compositional data analysis methods exhibit mixed results. Transformation methods, such as Blom and NPN, demonstrate promise in capturing complex associations. Batch correction methods, including BMC and Limma, consistently outperform other approaches. However, the influence of normalization methods is constrained by population effects, disease effects, and batch effects. These results provide insights for selecting appropriate normalization approaches in microbiome research, improving predictive models, and advancing personalized medicine. Future research should explore larger and more diverse datasets and develop tailored normalization strategies for microbiome data analysis.

The human microbiome is a complex ecosystem of microorganisms that exist in symbiosis with the human body[1]. Extensive research has established that the human microbiome plays crucial roles in numerous physiological processes, including digestion, metabolism, immune system modulation, and even cognitive functions. Disruptions in the delicate microbial balance, known as dysbiosis, have been linked to a wide range of health conditions, including obesity[2,3], diabetes[4], inflammatory bowel disease[5,6], allergies[7], and several types of cancer[8,9].

The advent of high-throughput sequencing technologies has revolutionized the field of microbiome research, enabling comprehensive profiling of microbial communities and providing insights into their roles in different physiological processes and disease states[10]. However, the analysis of microbiome data poses significant challenges due to inherent heterogeneity and variability across samples. Sources of variation can stem from technical differences in sequencing protocols[11], variations in sample collection[12] and processing methods[13], as well as biological diversity among individuals and populations. To extract meaningful insights from microbiome data, it is crucial to account for and mitigate these sources of variation.

Normalization methods have emerged as vital tools in addressing the heterogeneity and biases present in microbiome data. These methods aim to remove technical and biological biases, standardize data across samples, and enhance comparability between datasets. Various normalization approaches have been proposed, ranging from simple scaling methods to more advanced statistical techniques. Comparisons of normalization methods have been performed in the context of data distributions[14,15] and differential analysis[16–20]. Genotype-to-phenotype mapping is an essential problem in the current genomic era. In the realm of differential analysis and prediction,

[1]Frontier Science Center for Nonlinear Expectations, Ministry of Education, Qingdao 266237, China. [2]Research Center for Mathematics and Interdisciplinary Sciences, Shandong University, Qingdao 266237, China. [3]School of Mathematics, Shandong University, Jinan 250100, China. [4]Quantitative and Computational Biology Department, University of Southern California, Los Angeles 90089, USA. ✉email: yhluan@sdu.edu.cn

the application of normalization methods differs in their objectives. In differential analysis, the main objective of normalization among different datasets is to remove or mitigate spurious associations between microbes and diseases. On the other hand, the main objective of normalization for phenotype prediction is to increase prediction accuracy, robustness, reliability and generalizability of the trained model to the unseen testing data. However, the impact of normalization methods on phenotype predictions mainly focused on DNA microarray data and RNA-Seq data. Zwiener et al.[21] found rank-based transformations performed well in all scenarios in real RNA-Seq datasets. Franks et al.[22] proposed feature-wise quantile normalization (FSQN) and found FSQN successfully removes platform-based bias from RNA-Seq data, regardless of feature scaling or machine learning algorithm. Given the central role of normalization in microbiome data analysis and the lack of current methods comparison for microbiome data, there is a need to systematically evaluate their performance, particularly in the context of disease prediction.

In this paper, we provide a review of existing normalization methods and present a comprehensive evaluation of various normalization methods in predicting binary phenotypes using microbiome data. We examine the performance of scaling methods, compositional data analysis methods, transformation methods, and batch correction methods across simulated datasets and real datasets. Our analysis includes an assessment of prediction accuracy using metrics such as the area under the receiver operating characteristic curve (AUC), prediction accuracy, sensitivity, specificity, and the rank ordering of different methods.

By comparing and contrasting the performance of normalization methods across different datasets and phenotypic outcomes, we aim to provide insights into the strengths and limitations of each approach. This research will assist researchers and practitioners in selecting appropriate normalization methods for microbiome data analysis, thereby enhancing the robustness and reliability of predictive models in microbiome research.

Results

Different datasets have different background distributions

There are eight publicly accessible colorectal cancer (CRC) datasets shown in Table 1, including Feng[25], Gupta[26,68], Thomas[8], Vogtmann[28], Wirbel[29], Yachida[30], Yu[9], and Zeller[31]. In total, we included 1260 samples (625 controls, 635 CRC cases) from multiple countries such as the USA, China, France, etc. The participant demographics ranged from 21 to 90 years, with a male representation of 59.6%. The datasets were characterized by diverse body mass index (BMI) values and included subjects with other health conditions such as hypertension, hypercholesterolemia, and Type 2 Diabetes (T2D). DNA extraction and sequencing were conducted using various protocols and platforms. Our analysis aimed to examine the background distribution differences among these datasets.

In order to assess population differences across the CRC datasets, a PCoA plot based on Bray Curtis distance was generated. Figure 1a revealed distinct separations between different datasets, suggesting variations in microbial composition among the populations. Although the observed separation accounted for a small proportion (7.9%) of the total variance, statistical significance was confirmed through the PERMANOVA test ($p = 0.001$). These findings underscored the substantial heterogeneity in microbial communities across diverse CRC datasets, despite the relatively modest contribution to the overall variance. To quantify the overlaps of these datasets, we computed the average Bray-Curtis distance (Fig.1b). The dispersion of individual datasets was represented on the diagonal, with the largest dispersion observed in the Gupta dataset. Among the off-diagonal values that measured the average distance between samples in different datasets, Feng and Gupta exhibited the lowest overlap, with a distance of 0.901. Consequently, controls from these two datasets were selected as the template data for subsequent simulations in scenario 1. Mixing these two populations with decided proportions allowed us to control the heterogeneities between simulated populations.

Our analysis also extended to five distinct IBD datasets, as depicted in supplementary Table S1. These included the Hall[32], HMP[5,70], Ijaz[33], Nielsen[35], and Vila[6] datasets. Similar to the CRC datasets, the IBD datasets exhibited variations in geographical origin, age, BMI, and sequencing platforms. Supplementary Figure S1 revealed a clear separation between the different datasets (Supplementary Figure S1(**a**)) along with evident dataset dispersion variations (Supplementary Figure S1(**b**)). These observations underscore the fact that distinctive populations

Dataset	Country	No. of control	No. of CRC	No. of species	Zero percentage	DNA-Exk	Seq-Plat	Reference
Feng	Austria	61	46	578	80.0%	MoBio	IlluminaHiSeq	25
Gupta	Indian	30	30	308	83.6%	Qiagen	IlluminaNextSeq	26,68
Thomas	Italy	52	61	584	83.6%	Qiagen	IlluminaHiSeq	8
Vogtmann	United States of America	52	52	539	78.7%	Gnome	IlluminaHiSeq	28
Wirbel	Germany	65	60	537	80.8%	Gnome	IlluminaHiSeq	29
Yachida	Japan	251	258	697	87.3%	NA	IlluminaHiSeq	30
Yu	China	53	75	575	80.7%	Qiagen	IlluminaHiSeq	9
Zeller	France	61	53	629	81.0%	Gnome	IlluminaHiSeq	31

Table 1. Characteristics of CRC datasets, including country, number of control samples (No. of control), number of CRC samples (No. of CRC), number of species in each dataset (No. of species), percentage of zero values in each dataset (zero percentage), DNA extraction kits (DNA-Exk), sequencing platforms (Seq-Plat), and reference.

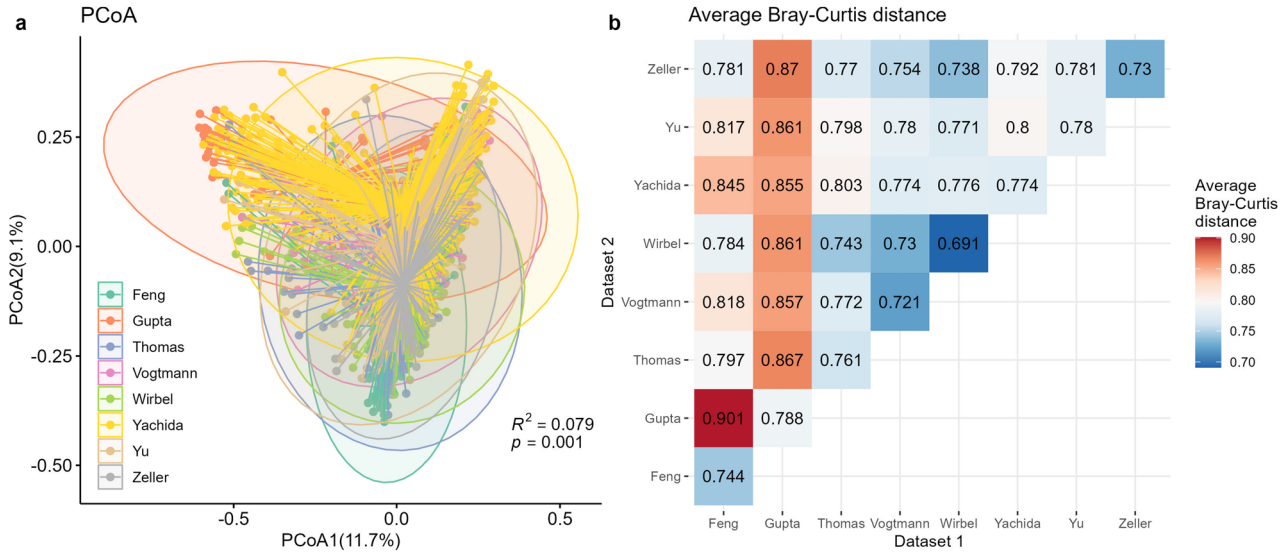

Figure 1. Different CRC populations had different background distribution patterns. (**a**) PCoA plot based on Bray-Curtis distance, with colors for different datasets. The variance explained by populations (PERMANOVA R^2) and its significance (PERMANOVA p value) were annotated in the figure. (**b**) Average Bray-Curtis distances between pairs of CRC datasets. Values on the diagonal referred to average Bray-Curtis distances between samples within the same dataset. Off-diagonal values refer to average Bray-Curtis distances between pairs of samples in different datasets. Larger values indicated a more dispersed distribution (on-diagonal) or bigger differences (off-diagonal). The figures were generated using R version 4.3.0.

are inherently marked by their unique background distributions, a factor that must be judiciously accounted for in any microbiome-related analysis.

Transformation and batch correction methods could enhance prediction performance for heterogeneous populations

In Scenario 1, the effects of different normalization methods on the prediction of binary phenotypes across diverse background distributions of taxa were investigated. The figures, including Figure 2, Supplementary Figures S2, S3, and S4, display the average performance metrics of 100 iterations: average AUC, accuracy, specificity, and sensitivity. Each panel in these figures represents a distinct disease effect, with each column denoting a population effect and rows indicating normalization methods.

When there were no population effects between the training and testing datasets ($ep = 0$), all normalization methods exhibited satisfactory performance, with average AUC, accuracy, sensitivity, and specificity values consistently achieving the maximum value of 1. However, as the population effects increased or disease effects decreased, an evident decline in these values was observed.

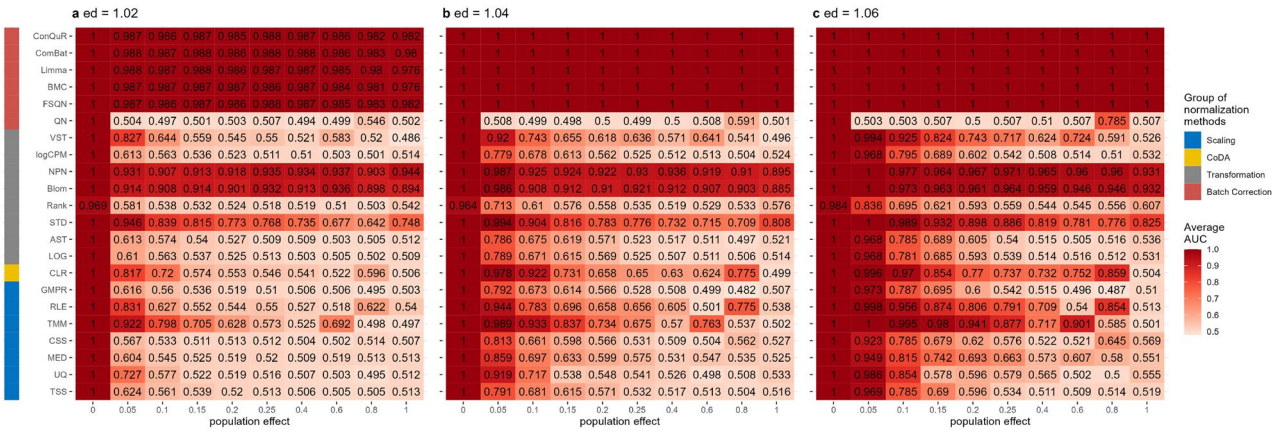

Figure 2. Heatmaps depicting average AUC values obtained from abundance profiles normalized by various methods for predicting simulated cases and controls in Scenario 1. The panels (**a**), (**b**), and (**c**) correspond to disease effects of 1.02, 1.04, and 1.06 respectively. The columns represent different values of population effects, while the rows represent different normalization methods, grouped based on their classifications in the left column. The figures were generated using R version 4.3.0.

When the differences between case and control were small (Figure 2(**a**), $ed = 1.02$), the prediction AUC values of scaling methods rapidly declined to 0.5 (random prediction value) as *ep* increased. TMM and RLE demonstrated better performances than TSS-based methods, such as UQ, MED, and CSS, in a wider range of conditions. Notably, TMM maintained an AUC value above 0.6 when $ep < 0.2$. As disease effects increased (Figure 2(**b**) $ed = 1.04$ and (**c**) $ed = 1.06$), both TMM and RLE exhibited superior ability to remove sample differences for predictions compared to TSS-based methods. Regarding prediction accuracy, TMM sustained accuracy above 0.6 with $ed > 1.04$ and $ep < 0.1$, surpassing the accuracy of other techniques (Supplementary Figure S2). In comparison to TMM, the other normalization methods specifically designed for RNA-Seq data, such as RLE, showed a tendency to misclassify controls as cases in predictions. This resulted in a sensitivity close to 1 (Supplementary Figure S3) and a specificity close to 0 (Supplementary Figure S4) in scenarios with population effects between training and testing datasets ($ep > 0$). Similar outcomes were observed for TSS but not for TSS-based methods such as UQ, MED, and CSS.

While normalized counts are commonly used for analyzing microbiome data, they still exhibit skewed distributions, unequal variances, and extreme values, which may limit their effectiveness in situations with significant heterogeneity. To enhance cross-population prediction performance, we applied various commonly used transformations, including CLR, LOG, AST, STD, Rank, Blom, NPN, logCPM, and VST. These transformation methods aimed to address one or several problems. For instance, logCPM and LOG transformations resolved skewness and extreme values, STD focused on unequal variances, VST tackled unequal variances and extreme values, and AST, CLR, Rank, Blom, and NPN addressed all three issues. The yellow and grey bars in Figure 2 represent the average prediction AUC values obtained using abundance profiles transformed by different methods. LOG, AST, Rank, and logCPM showed performances similar to TSS, indicating a failure in distribution adjustment. Conversely, transformation methods that achieved data normality, such as Blom and NPN, effectively aligned the data distributions across different populations for both population effects (*ep*) and disease effects (*ed*). Additionally, STD generally improved prediction AUC values, while the performance of CLR and VST transformation decreased with increasing population effects (*ep*). However, the sensitivity of all transformation methods was close to 1 (Supplementary Figure S3), and the specificity was close to 0 (Supplementary Figure S4) in circumstances where $ep > 0$. Consequently, prediction accuracies remained around 0.5 (Supplementary Figure S2), even for methods like Blom, NPN, and STD that exhibited higher AUC values.

Surprisingly, the batch correction methods highlighted in red bars yielded promising prediction results with high AUC (Figure 2), accuracy (Supplementary Figure S3), sensitivity (Supplementary Figure S4), and specificity (Supplementary Figure S5), except for QN. QN forced the distribution of each sample to be the same, potentially distorting the true biological variation between case and control samples, making it difficult for the classifier to distinguish between the groups. This was also validated by its high sensitivity (Supplementary Figure S3) and low specificity (Supplementary Figure S4) values. While QN was only effective when the two populations originated from the same distribution, FSQN, BMC, limma, ComBat, and ConQuR significantly enhanced the reproducibility of response predictions, remaining unaffected by disease effects and population effects.

Batch correction methods can successfully remove batch effects within the same population

In Scenario 2, we examined studies within the same population that exhibited technical variations and differences across batches. These batch effects can lead to substantial heterogeneity among the data batches[71]. Figures 3, S5, S6, and S7, respectively, showed the average AUC, accuracy, sensitivity, and specificity values obtained from random forest models using abundance profiles normalized by various methods across 100 runs. Overall, all

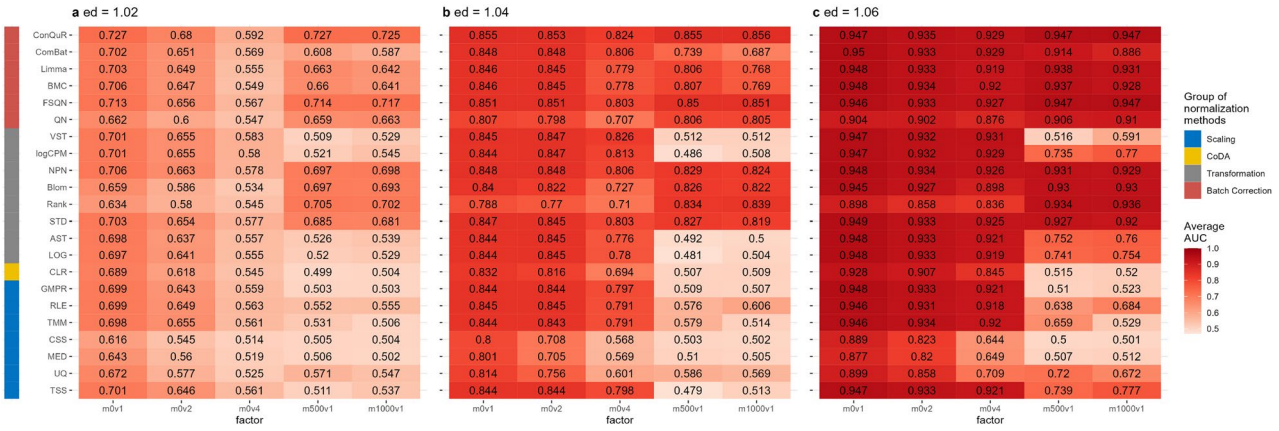

Figure 3. Heatmaps depicting average AUC values obtained from abundance profiles normalized by various methods for predicting simulated cases and controls in Scenario 2. The panels (**a**), (**b**), and (**c**) correspond to disease effects of 1.02, 1.04, and 1.06 respectively. The columns represent different combinations of batch mean and batch variation, with "m" for batch mean adjusting the mean and "v" for batch variance adjusting the variance. The rows represent different normalization methods, grouped based on their classifications in the left column. The figures were generated using R version 4.3.0.

these values demonstrated an upward trend with increasing disease effects. However, the normalization methods exhibited varying responses to changes in batch means and variances.

Figure 3a displayed the results obtained with disease effect equal to 1.02. When the batch variance remained fixed ($sev_{var} = 1$), pronounced response to additive batch means ($sev_{mean} = 0, 500, 1000$) was observed among the scaling methods and some transformation methods (CLR, LOG, AST, logCPM, VST). These methods exhibited a decrease in AUC scores from approximately 0.7 to around 0.5 when $sev_{mean} \neq 0$. In contrast, the STD, Rank, Blom, NPN, and all batch correction methods maintained a more robust level of AUC values (around 0.7) in the presence of varying batch means, as long as the batch variances did not change. These trends persisted with increasing disease effects, as depicted in Fig. 3b, c. Notably, among the methods more sensitive to batch means, scaling methods such as TMM and RLE exhibited a slight improvement in predictive accuracy as the batch means increased. Transformation methods like LOG, AST, and logCPM performed similarly.

The effects of batch variances on binary phenotype prediction remained consistent across different normalization methods. In Fig. 3a, when the batch mean was fixed at 0 and the batch variances were adjusted from 1 to 4, all normalization methods experienced an average decrease in AUC values of approximately 0.1. Among the scaling methods, namely MED, UQ, and CSS, which modified the scaling factor from TSS, consistently yielded lower AUC values compared to other methods for different batch variances ($sev_{var} = 1, 2, 4$). In Fig. 3c, with $ed = 1.06$, the influence of increased batch variance on prediction accuracy was reduced, indicating the dominance of disease effect in prediction. Most normalization methods achieved AUC scores above 0.9 when $sev_{var} = 4$, indicating successful removal of batch effects for predictions. Nonetheless, MED, UQ, and CSS continued to exhibit inferior ability in removing batch effects compared to other methods.

In scenario 2, the general trends of prediction accuracy (Supplementary Figure S5), sensitivity (Supplementary Figure S6), and specificity (Supplementary Figure S7) aligned with AUC values. It is noteworthy that ComBat maintained prediction accuracy, sensitivity, and specificity at a lower level than other batch correction methods when the batch variance remained constant and the batch mean increased, highlighting its limitations in addressing batch mean discrepancies.

The impact of disease model can be reduced by disease effects

In Scenario 3, we explored the influence of differences in disease models between the training and testing data on the prediction AUC scores. The results are presented in Figures 4, S8, S9, and S10. The overall trends in the relative performance of different normalization methods were consistent with the previous two scenarios. The AUC scores increased as the disease effects increased. And as expected, the AUC scores also increased as the number of overlapping disease-related taxa increased. For example, when $ed = 1.02$ (Fig. 4a), the AUC values obtained using abundance profiles normalized by different methods were all approximately 0.6 when there were 2 overlapping disease-associated taxa between the training and testing data. When the number of disease-associated taxa increased to 10, the optimal AUC scores increased to 0.7. The same pattern was observed with $ed = 1.04$ and $ed = 1.06$. When the disease effects increased to 1.06 (Fig. 4c), the majority of normalization methods achieved AUC scores exceeding 0.8, even when there were only 2 overlapped disease-associated taxa. This indicates that the impact of the disease model can be mitigated by stronger disease effects.

Figure 4 also illustrated that among the normalization methods we compared, scaling methods such as UQ, MED, and CSS had lower AUC values compared to other methods, as observed in the other two scenarios. QN also exhibited lower prediction performances. The other methods showed similar prediction performances with respect to different disease effects and different numbers of disease-associated taxa.

Supplementary Figures S8, S9, and S10 demonstrated a similar prediction performance of normalization methods measured by accuracy, sensitivity, and specificity.

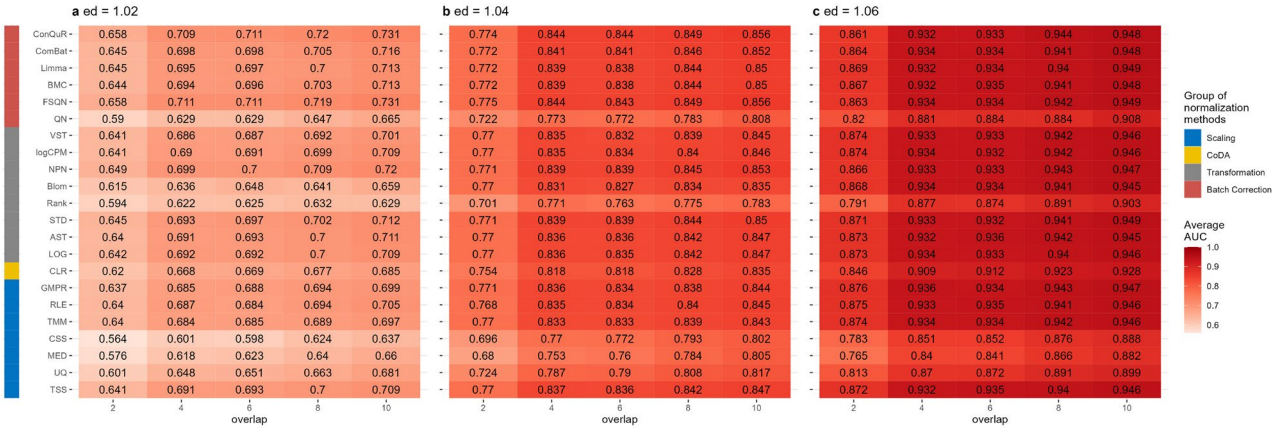

Figure 4. Heatmaps depicting average AUC values obtained from abundance profiles normalized by various methods for predicting simulated cases and controls in Scenario 3. The panels (**a**), (**b**), and (**c**) correspond to disease effects of 1.02, 1.04, and 1.06 respectively. The columns represent different numbers of overlapping disease-associated taxa in the training and testing datasets. The rows represent different normalization methods, grouped based on their classifications in the left column. The figures were generated using R version 4.3.0.

Batch correction methods are necessary for cross-dataset predictions

We next evaluate various normalization methods using 8 gut microbiome datasets from shotgun sequencing related to CRC (Table 1). These experimental datasets were retrieved from the R package curatedMetagenomicData with a sample size larger than 30 for either cases or controls. Datasets were paired with one for model training and the other for validation. For each method, the AUC score, accuracy, sensitivity, and specificity based on the normalized abundance using random forest was calculated. We repeated the predictions 30 times to account for the randomness of the prediction model and the average of the these values was reported for each study.

Supplementary Figure S11 presents box plots showing the AUC values obtained from the 30 repeated predictions. We observed unstable AUC values for most normalization methods when trained or tested on the Gupta dataset. This observation aligns with the data distribution depicted in Fig. 1, where Gupta exhibited the greatest dissimilarities and variability compared to other datasets. The same observation holds true for the Feng dataset. Overall, none of the normalization methods consistently improved the prediction AUC values to a specific level. The prediction accuracy remained dependent on both biological and technical factors. For example, when the model was trained on Gupta and tested on Feng, most methods yielded average AUC scores around 0.7, except for Rank and VST (Supplementary Figure S11(**b1**)). None of the normalization methods achieved an AUC value above 0.8 to significantly improve prediction performance.

The box plots of prediction accuracy, sensitivity, and specificity (Supplementary Figures S12, S13, and S14) are consistent with the results of AUC values, indicating that prediction outcomes are influenced by multiple factors, and normalization methods cannot fundamentally address the impact of heterogeneity on prediction reproducibility. Additionally, consistent with our observations in simulations, most methods exhibit a trend of high sensitivity and low specificity, suggesting that healthy individuals may be classified as diseased in clinical applications, requiring additional information for further assessment.

To quantify the performance of normalization methods, we ranked all normalization methods according to their average AUC, accuracy, sensitivity, and specificity values derived from models trained and tested on the same pair of training and testing datasets. The distributions of their ranks for each method are depicted in Figure 5. A higher ranking (lower values in the box plot) indicates a better prediction performance. Among the twenty-two normalization methods we compared, batch correction methods, including FSQN, BMC, and Limma, tended to have higher AUC values (Figure 5a) and higher accuracy (Figure 5b) than other methods. In comparison to FSQN, BMC and Limma exhibited a superior balance between sensitivity and specificity. Most transformation methods encountered an issue of high sensitivity but low specificity, particularly evident in STD, Rank, and QN (Fig. 5c, d). Scaling methods ranked behind batch correction methods and performed similarly to each other in CRC dataset predictions, indicating relatively small population effects in CRC datasets.

We also applied the normalization methods to IBD datasets listed in Supplementary Table S1 and conducted cross-dataset predictions. Supplementary Figures S15, S16, S17, and S18 illustrates the box plots of the AUC, accuracy, sensitivity, and specificity values obtained from 30 repeated predictions, respectively. And Supplementary Figure S19 visualizes the rank distributions for each method within pairs of IBD datasets. The results obtained were similar to those observed in the CRC dataset predictions. Among all the normalization methods, batch correction methods, including BMC and Limma, consistently demonstrated the best performance. Scaling methods, such as TMM, followed closely behind. However, FSQN exhibited variable performance, occasionally achieving good results while sometimes yielding poor results. Overall, the trends in IBD dataset predictions were consistent with the observations made in CRC dataset predictions.

Discussion

In our study, we considered three sources of heterogeneity between datasets: population effects, batch effects, and disease models. Population effect refers to variations arising from differences in population characteristics, including environmental factors, geographical locations, diet, and other population-specific features. If there are marked differences in the microbiome composition between the training and testing datasets, the trained model may struggle to distinguish disease-related microbiome patterns from population-specific variations. Batch effect arises from technical variations introduced during data collection or processing, such as sequencing technologies, sample preparation, or other experimental procedures. These batch effects may confound the true microbial signatures associated with the disease status, resulting in diminished generalization performance. Disease model represents the underlying patterns and features associated with the disease phenotype, and disparities in this regard can lead to decreased predictive performance, as a model trained on one dataset may encounter difficulties in generalizing effectively to another dataset. We conducted a comprehensive evaluation of various normalization methods for predicting binary phenotypes with the impact of heterogeneity from different sources. The results revealed important insights into the performance and suitability of different normalization approaches in the context of disease prediction.

Our findings demonstrated that no single normalization method consistently outperformed others across all datasets and phenotypic outcomes. This suggests that the choice of normalization method should be carefully considered based on the specific dataset characteristics and research objectives. However, certain trends and patterns did emerge from our analysis.

Among the scaling methods, methods such as TMM performed comparably well, indicating their effectiveness in reducing technical variations and improving the comparability of data across samples. These methods are relatively simple and straightforward to implement, making them practical choices for normalization in microbiome data analysis.

Interestingly, compositional data analysis methods, CLR, exhibited mixed performance across different datasets. While it has been widely used in microbial community analysis, our results suggest that its effectiveness in disease prediction may vary depending on the specific dataset and phenotypic outcome. Further investigation

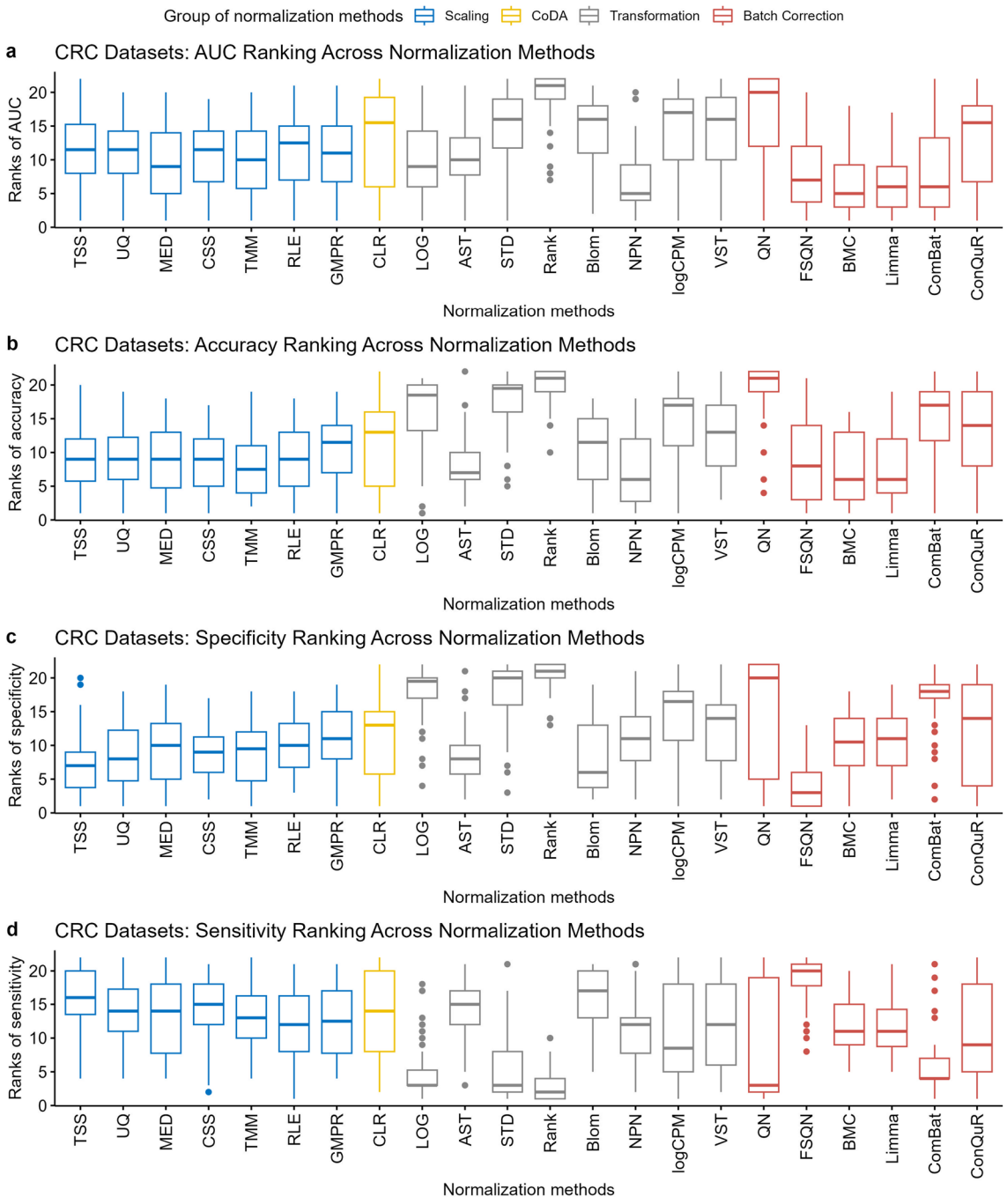

Figure 5. Distribution of ranks for 22 normalization methods in cross-dataset prediction on CRC datasets. The normalization methods are ranked based on the average AUC (**a**), average accuracy (**b**), average sensitivity (**c**), and average specificity (**d**) under the same pair of training and testing datasets. The figures were generated using R version 4.3.0.

is needed to understand the underlying factors influencing the performance of compositional data analysis methods in predicting binary phenotypes.

Transformation methods, including NPN and Blom, showed promising results in some datasets according to the prediction AUC values, highlighting their potential to improve prediction performance by capturing nonlinear relationships and addressing skewed distributions. These methods offer flexibility in handling diverse data types and can be particularly valuable in situations where data transformation is necessary to meet model assumptions. However, trade-offs need to be made between prediction sensitivity and specificity when appluying transformation methods.

Batch correction methods, such as BMC and Limma, consistently performed well across multiple datasets. These methods effectively accounted for batch effects, which are often present in multi-center or multi-cohort studies. The ability to remove batch effects is critical in ensuring accurate and reliable predictions, especially

when integrating data from different sources. Based on our findings, we recommend incorporating batch correction methods in cross-dataset binary phenotype prediction of metagenomic data. This involves utilizing scaling methods to mitigate biases attributed to sequencing technology, followed by LOG transformation to approximate a more normally distributed data, aligning with the assumptions of batch correction methods. By subsequently applying batch correction methods, we enhance the robustness of the analysis. We believe that this pipeline can improve the accuracy and reliability of phenotype cross-dataset predictions based on metagenomic data.

It is worth noting that the performance of normalization methods was influenced by the heterogeneity of the datasets. The relative impact of heterogeneity from different sources depends on the nature of the data and the extent of variation in each factor. For instance, if the population effect is pronounced and not adequately controlled, the model might capture population-specific differences instead of disease-related patterns. Likewise, if batch effects are left unaddressed, the model may overfit on technical variations instead of discerning true biological signals associated with the case-control status. In datasets where there were substantial biological and technical variations, the prediction accuracy remained primarily determined by these factors rather than the choice of normalization method. This emphasizes that proper preprocessing, normalization, and consideration of potential confounders are essential for building robust and generalizable predictive models.

Overall, our study underscores the need for careful consideration and evaluation of normalization methods in microbiome data analysis, particularly in the context of disease prediction. Researchers and practitioners should take into account the specific characteristics of their datasets, including population heterogeneity, disease effects, and technical variations when selecting and applying normalization methods. Additionally, future research should focus on developing novel normalization approaches that are tailored to the unique challenges of microbiome data and explore their performance in larger and more diverse datasets.

In conclusion, our comprehensive evaluation of normalization methods provides valuable insights into their performance in predicting binary phenotypes using microbiome data. This research contributes to the advancement of robust and reliable methodologies in microbiome research and paves the way for more accurate disease prediction and personalized therapeutic interventions based on the human microbiome.

Materials and methods

Real metagenomic dataests

As the first application example, we analyzed shotgun sequencing data from patients with colorectal cancer (CRC) obtained from the R package *curatedMetagenomicData* v3.8.0[23]. The taxonomic profiles for each dataset were determined using MetaPhlAn3[24], which ensures consistency in downstream analysis. A total of nine CRC datasets are available[8,9,25–31]. We excluded studies with sample sizes of less than 30 for either cases or controls, resulting in eight accessible CRC datasets for our analysis. A detailed summary outlining the distinctive characteristics of these eight CRC datasets can be found in Table 1.

As the second application example, we analyzed shotgun sequencing data from patients with inflammatory bowel disease (IBD) from the R package curatedMetagenomicData v3.8.0[23]. There are 6 available IBD datasets in *curatedMetagenomicData*[5,6,32–35]. Similarly to the CRC datasets, we excluded studies with sample sizes less than 30 for either cases or controls from the analysis. A summary of the characteristics of the IBD datasets can be found in Supplementary Table S1.

Statistical analysis

We calculated the microbial relative abundance for each sample and used the Bray-Curtis distance[36] to compare the dissimilarities between samples. This distance was computed using the function *vegdist()* from R package *vegan*[37]. To visualize the clustering of samples effectively, we performed principal coordinate analysis (PCoA) through the *pcoa()* function from R package *ape*[38]. To assess the variance attributable to datasets, we conducted the permutational multivariate analysis of variance (PERMANOVA)[39] with *adonis()* function in R package *vegan*[37].

Normalization methods

A number of normalization methods could be applied to microbiome data for data analyses. For the purpose of predicting the unknown disease status of samples, we try to transform or normalize our data to satisfy the assumption that training and testing data are drawn from the same distribution. Seven scaling methods, one compositional data analysis method, eight transformation methods, and six batch correction methods were compared in this analysis. Our study is also the largest comparison in terms of prediction up to date according to our best knowledge.

Assume we have a dataset consisting of n samples and m features. Denote c_{ij} as the count for taxon i in sample j. With this notation, the steps and formula of normalization methods can be briefly introduced as follows.

Scaling methods

A commonly used method for normalizing microbiome data is scaling. Its basic idea is to divide counts in the taxa count table by a scaling factor or normalization factor to remove biases resulting from sequencing technology:

$$x_{ij} = \frac{c_{ij}}{s_j},$$

where x_{ij} is the normalized abundance for taxon i in sample j, s_j is the scaling/normalization factor for sample j. We investigated seven popular scaling methods (Table 2) in our analysis, including TSS, UQ, MED, CSS in *metagenomeSeq*, TMM in *edgeR*, RLE in *DESeq2*, and GMPR in *GUniFrac*.

	Methods	Data designed for	Preprocessing	R function	R package
Scaling methods	TSS	/	None	sum()	stats
	UQ	RNA-Seq	None	quantile()	stats
	MED	RNA-Seq	None	median()	stats
	CSS	microbiome	None	cumNorm()	metagenomeSeq
	TMM	RNA-Seq	None	calcNormFactors()	edgeR
	RLE	RNA-Seq	None	estimateSizeFactors()	DESeq2
	GMPR	microbiome	None	GMPR()	GUniFrac
CoDA	CLR	TSS	TSS	clr()	compositions
Transformation methods	LOG	/	TSS	log()	stats
	AST	/	TSS	asin(), sqrt()	stats
	STD	/	TSS	center(), scale()	stats
	Rank	RNA-Seq	TSS	rank()	stats
	Blom	RNA-Seq	TSS	qnorm(), rank()	stats
	NPN	/	TSS	huge.npn()	huge
	logCPM	RNA-Seq	None	cpm()	edgeR
	VST	RNA-Seq	None	varianceStabilizingTransformation()	DESeq2
Batch correction methods	QN	DNA microarray	TSS, LOG	normalize.quantiles.use.target()	preprocessCore
	FSQN	RNA-Seq	TSS, LOG	quantileNormalizeByFeature()	FSQN
	BMC	DNA microarray	TSS, LOG	pamr.batchadjust()	pamr
	Limma	DNA microarray	TSS, LOG	removeBatchEffect()	limma
	ComBat	DNA microarray	TSS, LOG	ComBat()	sva
	ConQuR	microbiome	None	ConQuR()	conqur

Table 2. Summary of normalization methods, including seven scaling methods, one compositional data analysis (CoDA) method, eight transformation methods, and six batch correction methods.

Total Sum Scaling (TSS)[14]: Counts are divided by the total number of reads in that sample.

$$s_j^{\mathrm{TSS}} = \sum_i c_{ij}. \tag{1}$$

Upper Quartile (UQ)[14,40]: Similar to TSS, it scales each sample by the upper quartile of counts different from 0 in that sample.

$$s_j^{\mathrm{UQ}} = q^3(P_j), \tag{2}$$

where $q^3(\cdot)$ is the function of estimating upper quartile, and $P_j = \{c_{ij} | c_{ij} > 0, i = 1, \cdots, n\}$ represents a set of counts different from 0 in sample j.

Median (MED)[14]: Also similar to TSS, the total number of reads is replaced by the median counts different from 0 in the computation of the scaling factor.

$$s_j^{\mathrm{MED}} = \mathrm{Median}\left(P_j\right), \tag{3}$$

where Median$(\cdot)$ is the function of estimating median, and $P_j = \{c_{ij} | c_{ij} > 0, i = 1, \cdots, n\}$ represents a set of counts different from 0 in sample j.

Cumulative Sum Scaling (CSS)[41]: CSS modified TSS for microbiome data in a sample-specific manner. It selects the scaling factor as the cumulative sum of counts, up to a percentile $\hat{l}$ determined by the data:

$$s_j^{\mathrm{CSS}} = \frac{\sum_{i|i \in M_j} c_{ij}}{N^{\mathrm{CSS}}}, \tag{4}$$

where $M_j = \{c_{ij} | c_{ij} \leq q_{\hat{l}}(c_j)\}$ denotes the taxa included in the cumulative summation for sample j, and N^{CSS} is an appropriately chosen normalization constant. This scaling method is implemented by calling the *cumNorm()* function in the R package *metagenomeSeq*[41].

Trimmed Mean of M-values (TMM)[42]: TMM is a popular normalization method for RNA-Seq data with the assumption that most genes are not differentially expressed. It selects a reference sample first and views the others as test samples. If not specified, the sample with count-per-million upper quantile closest to the mean upper quantile is set as the reference. The scale factor between the test sample and the reference sample is estimated by the ratio of two observed relative abundance for a taxon i. The log2 of the ratio is called M value, $M_{jk}^i = \log_2 \frac{c_{ij}/\sum_i c_{ij}}{c_{ik}/\sum_i c_{ik}}$, and the log2 of the geometric mean of the observed relative abundance is called A value,

$A^i_{jk} = \frac{1}{2}\log_2\left(\frac{c_{ij}}{\sum_i c_{ij}}\frac{c_{ik}}{\sum_i c_{ik}}\right)$. By default, it trims the M values by 30% and the A values by 5%. Then the weighted sum of M values can be used to calculate the scale factor of sample j to sample k:

$$\log_2\left(s^{\text{TMM}}_{jk}\right) = \frac{\sum_{i\in m^{\text{TMM}}_{jk}}\left(w^i_{jk}M^i_{jk}\right)}{\sum_{i\in m^{\text{TMM}}_{jk}}\left(w^i_{jk}\right)}, \tag{5}$$

where m^{TMM}_{jk} is the remaining taxa after the trimming step, and weight $w^i_{jk} = \frac{\sum_i c_{ij}-c_{ij}}{c_{ij}\sum_i c_{ij}} + \frac{\sum_i c_{ik}-c_{ik}}{c_{ik}\sum_i c_{ik}}$. This scaling method is implemented using *calcNormFactors()* function in the *edgeR*[43] Bioconductor package.

Relative log expression (RLE)[44]: RLE is another widely used method for RNA-Seq data and relies on the same assumption that there is a large invariant part in the count data. It first calculates the geometric mean of the counts to a gene from all the samples and then computes the ratio of a raw count over the geometric mean to the same gene. The scale factor of a sample is obtained as the median of the ratios for the sample:

$$s^{\text{RLE}}_j = \text{Median}_i\left\{\frac{c_{ij}}{G(c_i)}\right\}, \tag{6}$$

where $G(c_i) = \left(\prod_{j=1}^m c_{ij}\right)^{\frac{1}{m}}$ is the geometric mean of gene i. By setting the *type="poscounts"* of *estimateSizeFactors()* function in the *DESeq2*[45] Bioconductor package, a modified geometric mean is computed. This calculation takes the n-th root of the product of the non-zero counts to deal with zeros in microbiome data.

Geometric mean of pairwise ratios (GMPR)[46]: GMPR extends the idea of RLE normalization by reversing the order of computing geometric and median to overcome the zero inflation problem in microbiome data. The scale factor for a given sample j using reference sample k is calculated as

$$s^{\text{GMPR}}_j = \left(\prod_j \text{Median}_{i|c_{ij}\cdot c_{ik}\neq 0}\left\{\frac{c_{ij}}{c_{ik}}\right\}\right)^{\frac{1}{m}}. \tag{7}$$

This scaling method is implemented using *GMPR()* function in the *GUniFrac*[47] package.

Compositional data analysis (CoDA) methods

Gloor et. al.[48] pointed out that microbiome datasets generated by high-throughput sequencing are compositional because they have an arbitrary total imposed by the instrument. Thus several methods were proposed to eliminate the effect of sampling fraction by converting the abundances to log ratios within each sample. These commonly used methods in compositional data analysis include additive log-ratio transformation (ALR)[49], centered log-ratio transformation (CLR)[49], and isometric log-ratio transformation (ILR)[49]. ALR and ILR convert n dimensional taxon vector to $n-1$ dimensional data in the Euclidean space, with the challenge of choosing a reference taxon. Due to the large number of taxa and the resulting computing problem, we only considered CLR in our analysis.

Centered Log-Ratio (CLR)[49]: CLR transformation is a compositional data transformation that takes the log-ratio of counts and their geometric means. This is done within each sample based on relative abundances. This can be written in mathematical form as:

$$\text{clr}(x_j) = \left[\log\frac{x_{1j}}{G(x_j)}, \cdots, \log\frac{x_{nj}}{G(x_j)}\right] \tag{8}$$

where x_{ij} is the relative abundance of taxon $i, i = 1, \cdots, n$ in sample $j, j = 1, \cdots, m$, $G(x_j) = \left(\prod_{i=1}^n x_{ij}\right)^{\frac{1}{n}}$ is the geometric mean of sample j with a pseudo count 0.65 times minimum non-zero abundance added to 0 values[50]. This transformation is implemented using *clr()* function in R package *compositions*[51].

Transformation methods

Microbiome data have problematic properties such as skewed distribution, unequal variances for the individual taxon, and extreme values. We propose to transform microbiome data before fitting the prediction model to handle either one, two, or all of these problems. Let c_{ij} and x_{ij} be the count and relative abundance of taxon $i, i = 1, \cdots, n$ in sample $j, j = 1, \cdots, m$, respectively. Table 2 gives a summary of transformation methods considered in this study, including LOG, AST, STD, Rank, Blom, NPN in huge, logCPM in *edgeR*, and VST in *DESeq2*.

LOG: Log transformation is often used for taxa with skewed distribution so that the transformed abundances are more or less normally distributed[21]. A pseudo count 0.65 times the minimum non-zero abundance is added to the zero values before log transformation to avoid infinite values[50].

$$\log(x_i) = \left(\log x_{i1}, \cdots, \log x_{im}\right). \tag{9}$$

Arcsine square-root (AST): AST transformed data have less extreme values compared to the untransformed data and are more or less normally distributed. It is defined as

$$\text{AST}(x_i) = \left(\arcsin\sqrt{x_{i1}}, \cdots, \arcsin\sqrt{x_{im}}\right). \tag{10}$$

Standardization (STD)[21]: STD is the default implementation in many regression analyses to reduce the variations of features (taxa in our analysis):

$$\mathrm{STD}(x_i) = \left(\frac{x_{i1} - \mu_i}{\sigma_i}, \cdots, \frac{x_{im} - \mu_i}{\sigma_i}\right), \tag{11}$$

where μ_i and σ_i is the mean and standard deviation of taxon i separately.

Rank[21]: Rank transformation is a simple and popular method used in non-parametric statistics. The rank-transformed features are uniformly distributed from zero to the sample size m. A small noise term $\epsilon_{ij} \sim N(0, 10^{-10})$ is added before data transformation to handle the ties of zero counts.

$$\mathrm{Rank}(x_i) = (r_{i1}, \cdots, r_{im}), \tag{12}$$

where $r_{ij}, j = 1, \cdots, m$ is the corresponding rank for relative abundance $x_{ij}, j = 1, \cdots, m$ in taxon i.

Blom[21,52]: Blom transformation is based on rank transformation. The uniformly distributed ranks are further transformed into a standard normal distribution:

$$\mathrm{Blom}(x_i) = \left(\Phi^{-1}\left(\frac{r_{i1} - c}{m + 1}\right), \cdots, \Phi^{-1}\left(\frac{r_{im} - c}{m + 1}\right)\right), \tag{13}$$

where $c = \frac{3}{8}$ is a constant, $\Phi^{-1}(\cdot)$ denotes the quantile function of normal distribution, and $r_{ij}, j = 1, \cdots, m$ is the corresponding rank for relative abundance $x_{ij}, j = 1, \cdots, m$ in taxon i.

Non-paranormal (NPN)[53]: NPN transformation is designed to be used as part of an improved graphical lasso that first transforms variables to univariate smooth functions that estimate a Gaussian copula. The transformation can also be used alone for analysis. Let Φ denote the Gaussian cumulative distribution function, then we can estimate the transformed data using

$$\mathrm{NPN}(x_{ij}) = \begin{cases} \Phi^{-1}(\delta), \text{ if } \hat{r}_{ij} \le \delta, \\ \Phi^{-1}(\hat{r}_{ij}), \text{ if } \delta < \hat{r}_{ij} \le 1 - \delta, \\ \Phi^{-1}(1 - \delta), \text{ if } \hat{r}_{ij} \ge 1 - \delta, \end{cases} \tag{14}$$

where $\hat{r}_{ij} = \frac{r_{ij}}{m+1}$, and $\delta = \frac{1}{4m^{1/4}\sqrt{\pi \log m}}$. This transformation is implemented using *huge.npn()* function in R package *huge*[54].

Log counts per million (logCPM): logCPM refers to the log counts per million, which is a useful descriptive measure for the expression level of a gene for RNA-Seq data. We applied it to the microbiome data. A pseudo count 0.65 times the minimum non-zero abundance is added to the zero values before log transformation.

$$\mathrm{logCPM}(c_i) = \left(\log_2 \frac{c_{i1}}{10^6}, \cdots, \log_2 \frac{c_{im}}{10^6}\right). \tag{15}$$

This transformation method is implemented using *cpm()* function in the *edgeR*[43] Bioconductor package.

Variance Stabilizing Transformation (VST)[44]: VST models the relationship between mean μ_i and variance σ_i^2 for each gene i:

$$\mathrm{VST}(c_{ij}) = \int_0^{c_{ij}} \frac{1}{v(\mu_i)} d\mu_i, \tag{16}$$

where $v(\mu_i) = \sigma_i^2 = \mu_i + a_i \mu_i^2$, with $a_i = a_0 + \frac{a_1}{\mu_i}$ being a dispersion parameter and a_0 and a_1 are estimated in a generalized linear model. A pseudo count 1 was added to zero values. This transformation is implemented using *varianceStabilizingTransformation()* function in the *DESeq2*[45] Bioconductor package.

Batch correction methods

Batch effects in many genomic technologies result from various specimen processing. And they often cannot be fully addressed by normalization methods alone. Many methods have been proposed to remove batch effects. Here we studied six commonly used approaches, including QN in *preprocessCore*, FSQN in *FSQN*, BMC in *pamr*, limma in *limma*, ComBat in *sva*, and ConQuR in *conqur* (Table 2).

Quantile normalization (QN)[55]: QN is initially developed for use with DNA microarrays, but has since been expanded to accommodate a wide range of data types, including microbiome data. Given a reference distribution, QN essentially replaces each value in a target distribution with the corresponding value from a reference distribution, based on identical rank order. In cases where the reference distribution encompasses multiple samples, the reference distribution should be first quantile normalized across all samples[56]. In our analysis, we designated the training data as the reference distribution. We applied QN to log-transformed relative abundances, substituting zeros with a pseudo count that was calculated as 0.65 times the minimum non-zero abundance across the entire abundance table. The reference distribution is obtained using function *normalize.quantiles.determine.target()* in R package *preprocessCore*[57]. And the batch effects are removed using function *normalize.quantiles.use.target()* in R package *preprocessCore*[57].

Feature specific quantile normalization (FSQN)[22]: FSQN is similar to QN, except for quantile normalizing the genes rather than samples. The reference distribution is the taxon in the training set and the target distribution is the taxon in the testing set. It is applied to log-transformed relative abundance data, with zeros replaced with

pseudo count 0.65 times the minimum non-zero abundance across the entire abundance table, using function *quantileNormalizeByFeature()* in R package *FSQN*[22].

Batch mean centering (BMC)[58]: BMC centers the data batch by batch. The mean abundance per taxon for a given dataset is subtracted from the individual taxon abundance. It is applied to log-transformed relative abundance data, with zeros replaced with pseudo count 0.65 times the minimum non-zero abundance across the entire abundance table, using *pamr.batchadjust()* function from *pamr* R package[59].

Linear models for microarray data (Limma)[60]: Limma fits a linear model to remove the batch effects. We first calculate the relative abundances and apply a log2 transformation to them. A pseudo count 0.65 times the minimum non-zero abundance across the entire abundance table was added to zeros to avoid infinite values for log transformation. The *removeBatchEffect()* function in R package *limma*[60] is then used to correct for batch effects, taking the log2 relative abundance data and batch information as inputs.

ComBat[61]: ComBat uses an empirical Bayes framework to estimate and remove the batch effects while preserving the biological variation of interest. Similar to Limma, the relative abundance of microbiome data (zero replaced with pseudo count 0.65 times the minimum none-zero abundance across the entire abundance table) was log-transformed prior to batch correction. This correction method is implemented using the function *ComBat()* in R package *sva*[62].

Conditional quantile regression (ConQuR)[63]: ConQuR conducts batch effects removal from a count table by conditional quantile regression. This batch correction method is implemented using function ConQuR in the R package ConQuR[63].

The random forest classifiers

In both the CRC and the IBD datasets, we aimed to predict whether a sample originated from a case subject (CRC/IBD) or a control subject.

The training and testing datasets underwent normalization to minimize heterogeneities both within and across datasets. For scaling methods that select references, such as TMM and RLE, and transformation methods that make prediction covariates (taxa) drawn from the same distribution, such as STD, Rank, Blom, NPN, and VST, we first normalized the training data. Then we combined the training and testing data together and normalized the combined data. Finally, we chose the samples from the testing data as the normalized testing data. This approach ensures the consistency in normalization of training and testing data[64].

We performed prediction of disease status using random forest, which has been shown to outperform other learning tools for most microbiome data[65]. The random forest models were implemented using function *train()* in R package *caret*[66] with 1,000 decision trees, and the number of variables at each decision tree was tuned using grid search by 10-fold cross-validation.

In the testing set, each sample was assigned a disease probability score. Initially, we adjusted the score threshold to calculate the True Positive Rate (TPR) and False Positive Rate (FPR) at varying thresholds and generated a Receiver Operating Characteristic (ROC) curve. The Area Under the ROC Curve (AUROC) was utilized as the metric for prediction accuracy evaluation. Subsequently, we set a fixed threshold at 0.5. Samples with a probability score exceeding this threshold were predicted as diseased (positive), while those below it were classified as non-diseased (negative). Measures such as accuracy, specificity, and sensitivity were computed to assess the prediction accuracy.

Simulation studies

A successful predictive model is transferable across datasets. To evaluate the impact of various normalization methods on binary phenotype prediction, we conducted simulations by creating two case-control populations, normalizing them using various methods, building prediction models with random forest on one simulated population, and testing them on the other in 3 different scenarios. The prediction accuracy, measured by AUC values, was evaluated for each of the 100 simulation runs in different scenarios.

Scenario 1: Different background distributions of taxa in populations

In the first scenario, we assumed that the heterogeneities between populations were due to variations in the background distributions of taxa, such as ethnicity or diet. McMurdie and Holmes[16] presented a way to simulate samples from different populations (Simulation A) and samples with case-control status (Simulation B) separately in such a scenario. In our simulations, we integrated these strategies and introduced certain modifications.

Our methodology began by determining the underlying taxon abundance levels for the training and testing populations. From Figure 1, the two least overlapping datasets, Gupta[26,68] and Feng[25], were chosen to be the template of training and testing sets, respectively. More specifically, 30 control samples and 183 species of the Gupta dataset were included for simulating the dataset for training, and 61 healthy samples and 468 species of the Feng dataset were included for simulating the dataset for testing. For each dataset, we had a count table with rows for taxa and columns for samples. Sum the rows to get the original vectors representing the underlying taxa abundance in different populations, denoted as $p_k, k = 1, 2$, respectively.

To investigate the impact of differences between two populations on cross-study prediction, we create pseudo-population vectors $v_k, k = 1, 2$:

$$v_1 = ep \cdot p_1 + (1 - ep) \cdot p_2,\ v_2 = p_2,$$

where ep is the population effect quantifying differences between two populations. Note that $v_1 - v_2 = ep \cdot (p_1 - p_2)$. Therefore, the differences between the two simulated populations increase with ep. At $ep = 0$, the two simulated populations share the same underlying distribution, resulting in no population differences between the training and testing datasets. Conversely, at $ep = 1$, the simulated populations exhibit

the largest possible differences. In our simulations, we examined the overall trend for different normalization methods by varying *ep* from 0 to 1 in increments of 0.2. For scaling methods and transformation methods that work effectively at smaller *ep* values, we set *ep* to range from 0 to 0.25 in increments of 0.05.

Out of the 154 shared taxa between the two populations, we randomly selected 10 taxa and hypothesized that these taxa were associated with a specific disease of interest. Considering that disease-associated taxa can either be enriched or depleted, we presumed the first 5 taxa to be enriched and the latter 5 to be depleted. These 10 taxa were fixed in the following analysis. The abundance vectors for simulated controls of selected disease-associated taxa were not changed ($v_k^{\text{ctrl}} = v_k, k = 1, 2$), while the abundance vectors for simulated cases of selected disease-associated taxa were defined as follows:

$$v_k^{\text{case}}[\text{enriched}] = v_k \cdot ed,\ v_k^{\text{case}}[\text{depleted}] = v_k/ed,\ k = 1, 2,$$

where $ed \in \{1.02, 1.04, 1.06\}$ denoted a disease effect factor that quantified the differences between cases and controls. As the value of *ed* increases, the difference between case and control samples becomes more marked. Once we had the new vectors, we re-normalized them into probability vectors denoted as $v_k^{\text{case}}, k = 1, 2$.

Pseudo probability for control sample *j* in population *k*, denoted as x_{kj}^{ctrl}, was generated under the assumption of a Dirichlet distribution: $x_{kj}^{\text{ctrl}} \sim Dir(\alpha_k^{\text{ctrl}})$, with $\alpha_k^{\text{ctrl}} = c \cdot v_k^{\text{ctrl}}$ for $k = 1, 2$. When *c* is very large, the variance of x_{kj}^{ctrl} will be close to 0, and it is similar to v_k^{ctrl}. To introduce some variability while generating non-zero probabilities, we set *c* to 1×10^6. The read counts for control sample *j* in population *k* was subsequently simulated using multinomial distribution, with a library size of 1, 000, 000, described by:

$$c_{kj}^{\text{ctrl}} \sim MN(\text{library size}, x_{kj}^{\text{ctrl}}),\ k = 1, 2.$$

The generation of case samples followed a similar procedure, with the creation of 50 control and 50 case samples within each population.

In the scenario where $ed = 1.02$ and $ep = 0$, both the training and testing datasets share the same background distribution. The proportion of zero values in the simulated training and testing sets is approximately 11.2% ± 0.2%. As *ep* increases, the background distribution in the testing dataset remains constant, resulting in a consistent proportion of zero values. In contrast, the proportion of zero values in the training dataset increases with the increase in *ep*. When $ep = 1$, the proportion of zero values reaches approximately 20% ± 0.2%. The value of *ed* does not affect the proportion of zero values in the training and testing data. Therefore, both $ed = 1.04$ and $ed = 1.06$ yield similar outcomes.

Scenario 2: Different batch effects in studies with the same background distribution of taxa in populations
In this scenario, we utilized Feng dataset[25] as the template for simulations. This ensured that the background distribution remained consistent between the training and testing datasets, thereby eliminating the population effects discussed in Scenario 1. We generated the read counts of training and testing data with 50 controls and 50 cases each by following the same procedure described in Scenario 1. It involved using multinomial distributions with a sample size of one million reads. The number of disease-associated taxa was set to 10 and disease effects varied from 1.02 to 1.06 with increments of 0.2.

To simulate batch effects, we followed a similar procedure as in Zhang et al[69]. They used the linear model assumed in the ComBat batch correction method[61] as the data-generating model for batch effects. Specifically, we assumed that both the mean (γ_{ik}) and variance (δ_{ik}) of taxon *i* were influenced by the batch *k*. The values of γ_{ik} and δ_{ik} were randomly drawn from normal and inverse gamma distributions:

$$\gamma_{ik} \sim N(\mu_k, \sigma_k^2),\ \delta_{ik} \sim \text{InvGamma}(\alpha_k, \beta_k).$$

To set the hyper-parameters ($\mu_k, \sigma_k, \alpha_k, \beta_k$), we specify two values to represent the severity of batch effects. This included three levels for batch effects on the mean ($sev_{mean} \in \{0, 500, 1000\}$) and three levels for batch effects on the variance ($sev_{var} \in \{1, 2, 4\}$). For each severity level, the variance of γ_{ik} and δ_{ik} was fixed at 0.01. The parameters are then added or multiplied to the expression mean and variance of the original study. The batch effects were only simulated on the training data while the testing dataset was unchanged.

In simulation scenario 2, where the background distribution remains consistent for both the training and testing sets, the proportion of zero values remains stable at 11.2% ± 0.2% in both datasets. However, when incorporating the batch mean into the expression mean, the proportion of zero values in the training data decreases to 0%. Conversely, when multiplying the batch variance with the expression variance, the proportion of zero values in the training data increases to 16% ± 0.2%.

Scenario 3: Different disease models of studies with the same background distribution of taxa in populations
In this scenario, we hypothesized that the model for disease-associated taxa could vary between populations. To avoid the population effects described in Scenario 1, we utilized the Feng dataset[25] as template for simulations. To avoid the batch effects described in Scenario 2, no batch effects were introduced into this simulation scenario.

For the selection of disease-associated taxa, we predefined 10 taxa for the training data. A subset of taxa was chosen from the initially selected 10 and additional taxa were included to maintain a total of 10 signature taxa in the testing data. The degree of similarity between the training and testing data was determined by the number of overlapping taxa, ranging from 2 to 10 with increments of 2. Subsequently, the two populations were simulated following the same procedure as in the previous two scenarios. The simulation parameters included 100 samples per population (50 controls and 50 cases), one million reads per sample, and a disease effect of 1.02, 1.04, 1.06.

In this scenario, both training and testing data share the same background distribution, and there are no batch effects. Therefore, the proportion of zero values in all count tables remains 11.2% ± 0.2%.

Data availability

All the CRC and IBD datasets used in this study are available in the R package curatedMetagenomicData (v3.8.0). All the codes used in the analysis can be found at https://github.com/wbb121/Norm-Methods-Comparison.

Received: 28 September 2023; Accepted: 20 March 2024
Published online: 25 March 2024

References

1. Ursell, L. K., Metcalf, J. L., Parfrey, L. W. & Knight, R. Defining the human microbiome. *Nutr. Rev.* **70**, S38–S44 (2012).
2. Ley, R. E. *et al.* Obesity alters gut microbial ecology. *Proc. Natl. Acad. Sci.* **102**, 11070–11075 (2005).
3. Ley, R. E., Turnbaugh, P. J., Klein, S. & Gordon, J. I. Human gut microbes associated with obesity. *Nature* **444**, 1022–1023 (2006).
4. Zhou, W. *et al.* Longitudinal multi-omics of host-microbe dynamics in prediabetes. *Nature* **569**, 663–671 (2019).
5. Lloyd-Price, J. *et al.* Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature* **569**, 655–662 (2019).
6. Vich Vila, A. et al. Gut microbiota composition and functional changes in inflammatory bowel disease and irritable bowel syndrome. Science translational medicine **10**, eaap8914 (2018).
7. Noverr, M. C. & Huffnagle, G. B. The 'microflora hypothesis' of allergic diseases. *Clin. Exp. Allergy* **35**, 1511–1520 (2005).
8. Thomas, A. M. *et al.* Metagenomic analysis of colorectal cancer datasets identifies cross-cohort microbial diagnostic signatures and a link with choline degradation. *Nat. Med.* **25**, 667–678 (2019).
9. Yu, J. *et al.* Metagenomic analysis of faecal microbiome as a tool towards targeted non-invasive biomarkers for colorectal cancer. *Gut* **66**, 70–78 (2017).
10. Wensel, C. R., Pluznick, J. L., Salzberg, S. L. & Sears, C. L. Next-generation sequencing: insights to advance clinical investigations of the microbiome. *J. Clin. Investig.* **132**, e154944 (2022).
11. D'Amore, R. *et al.* A comprehensive benchmarking study of protocols and sequencing platforms for 16s rrna community profiling. *BMC Genom.* **17**, 55 (2016).
12. Amir, A. *et al.* Correcting for microbial blooms in fecal samples during room-temperature shipping. *Msystems* **2**, e00199-16 (2017).
13. Bartolomaeus, T. U. *et al.* Quantifying technical confounders in microbiome studies. *Cardiovasc. Res.* **117**, 863–875 (2021).
14. Dillies, M.-A. *et al.* A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis. *Brief. Bioinform.* **14**, 671–683 (2013).
15. Müller, C. *et al.* Removing batch effects from longitudinal gene expression-quantile normalization plus combat as best approach for microarray transcriptome data. *PLoS ONE* **11**, e0156594 (2016).
16. McMurdie, P. J. & Holmes, S. Waste not, want not: why rarefying microbiome data is inadmissible. *PLoS Comput. Biol.* **10**, e1003531 (2014).
17. Weiss, S. *et al.* Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome* **5**, 1–18 (2017).
18. Du, R., An, L. & Fang, Z. Performance evaluation of normalization approaches for metagenomic compositional data on differential abundance analysis. New Frontiers of Biostatistics and Bioinformatics 329–344 (2018).
19. Gibbons, S. M., Duvallet, C. & Alm, E. J. Correcting for batch effects in case-control microbiome studies. *PLoS Comput. Biol.* **14**, e1006102 (2018).
20. Lin, H. & Peddada, S. D. Analysis of microbial compositions: a review of normalization and differential abundance analysis. *NPJ Biofilms Microbiomes* **6**, 60 (2020).
21. Zwiener, I., Frisch, B. & Binder, H. Transforming rna-seq data to improve the performance of prognostic gene signatures. *PLoS ONE* **9**, e85150 (2014).
22. Franks, J. M., Cai, G. & Whitfield, M. L. Feature specific quantile normalization enables cross-platform classification of molecular subtypes using gene expression data. *Bioinformatics* **34**, 1868–1874 (2018).
23. Pasolli, E. *et al.* Accessible, curated metagenomic data through experimenthub. *Nat. Methods* **14**, 1023–1024 (2017).
24. Beghini, F. et al. Integrating taxonomic, functional, and strain-level profiling of diverse microbial communities with biobakery 3. elife **10**, e65088 (2021).
25. Feng, Q. *et al.* Gut microbiome development along the colorectal adenoma-carcinoma sequence. *Nat. Commun.* **6**, 6528 (2015).
26. Gupta, A. *et al.* Association of flavonifractor plautii, a flavonoid-degrading bacterium, with the gut microbiome of colorectal cancer patients in india. *MSystems* **4**, e00438-19 (2019).
27. Hannigan, G. D., Duhaime, M. B., Ruffin, M. T. IV., Koumpouras, C. C. & Schloss, P. D. Diagnostic potential and interactive dynamics of the colorectal cancer virome. *MBio* **9**, e02248-18 (2018).
28. Vogtmann, E. *et al.* Colorectal cancer and the human gut microbiome: reproducibility with whole-genome shotgun sequencing. *PLoS ONE* **11**, e0155362 (2016).
29. Wirbel, J. *et al.* Meta-analysis of fecal metagenomes reveals global microbial signatures that are specific for colorectal cancer. *Nat. Med.* **25**, 679–689 (2019).
30. Yachida, S. *et al.* Metagenomic and metabolomic analyses reveal distinct stage-specific phenotypes of the gut microbiota in colorectal cancer. *Nat. Med.* **25**, 968–976 (2019).
31. Zeller, G. *et al.* Potential of fecal microbiota for early-stage detection of colorectal cancer. *Mol. Syst. Biol.* **10**, 766 (2014).
32. Hall, A. B. *et al.* A novel ruminococcus gnavus clade enriched in inflammatory bowel disease patients. *Genome Med.* **9**, 103 (2017).
33. Ijaz, U. Z. *et al.* The distinct features of microbial 'dysbiosis' of crohn's disease do not occur to the same extent in their unaffected, genetically-linked kindred. *PLoS ONE* **12**, e0172605 (2017).
34. Li, J. *et al.* An integrated catalog of reference genes in the human gut microbiome. *Nat. Biotechnol.* **32**, 834–841 (2014).
35. Nielsen, H. B. *et al.* Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. *Nat. Biotechnol.* **32**, 822–828 (2014).
36. Bray, J. R. & Curtis, J. T. An ordination of the upland forest communities of southern wisconsin. *Ecol. Monogr.* **27**, 326–349 (1957).
37. Oksanen, J. *et al.* Community ecology package. R package version 2.6-4 (2007).
38. Paradis, E. & Schliep, K. ape 5.0: an environment for modern phylogenetics and evolutionary analyses in r. Bioinformatics **35**, 526–528 (2019).
39. Anderson, M. J. A new method for non-parametric multivariate analysis of variance. *Austral Ecol.* **26**, 32–46 (2001).
40. Bullard, J. H., Purdom, E., Hansen, K. D. & Dudoit, S. Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. *BMC Bioinf.* **11**, 94 (2010).
41. Paulson, J. N., Stine, O. C., Bravo, H. C. & Pop, M. Differential abundance analysis for microbial marker-gene surveys. *Nat. Methods* **10**, 1200–1202 (2013).

42. Robinson, M. D. & Oshlack, A. A scaling normalization method for differential expression analysis of rna-seq data. *Genome Biol.* **11**, 2 (2010).
43. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edger: a bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics **26**, 139–140 (2010).
44. Anders, S. & Huber, W. Differential expression analysis for sequence count data. Nature Precedings 1–1 (2010).
45. Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome Biol.* **15**, 1–21 (2014).
46. Chen, L. *et al.* Gmpr: A robust normalization method for zero-inflated count data with application to microbiome sequencing data. *PeerJ* **6**, e4600 (2018).
47. Chen, J., Zhang, X. & Zhou, H. Gunifrac: generalized unifrac distances, distance-based multivariate methods and feature-based univariate methods for microbiome data analysis. R package version 1.7 (2018).
48. Gloor, G. B., Macklaim, J. M., Pawlowsky-Glahn, V. & Egozcue, J. J. Microbiome datasets are compositional: and this is not optional. *Front. Microbiol.* **8**, 2224 (2017).
49. Aitchison, J. The statistical analysis of compositional data. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **44**, 139–160 (1982).
50. Martín-Fernández, J. A., Barceló-Vidal, C. & Pawlowsky-Glahn, V. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Math. Geol.* **35**, 253–278 (2003).
51. Van den Boogaart, K. G. & Tolosana-Delgado, R. Compositions: a unified r package to analyze compositional data. *Comput. Geosci.* **34**, 320–338 (2008).
52. Beasley, T. M., Erickson, S. & Allison, D. B. Rank-based inverse normal transformations are increasingly used, but are they merited?. *Behav. Genet.* **39**, 580–595 (2009).
53. Liu, H., Lafferty, J. & Wasserman, L. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *J. Mach. Learn. Res.* **10**, 2295–2328 (2009).
54. Jiang, H. et al. huge: high-dimensional undirected graph estimation. R package version 1.3.5 (2021).
55. Bolstad, B. M., Irizarry, R. A., Åstrand, M. & Speed, T. P. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19**, 185–193 (2003).
56. Thompson, J. A., Tan, J. & Greene, C. S. Cross-platform normalization of microarray and rna-seq data for machine learning applications. *PeerJ* **4**, e1621 (2016).
57. Bolstad, B. M. preprocesscore: A collection of pre-processing functions. R package version 1.56.0 (2021).
58. Sims, A. H. *et al.* The removal of multiplicative, systematic bias allows integration of breast cancer gene expression datasets-improving meta-analysis and prediction of prognosis. *BMC Med. Genom.* **1**, 42 (2008).
59. Hastie, T., Tibshirani, R., Narasimhan, B. & Chu, G. Pam: prediction analysis for microarrays. R package version 1.56.1 **1**, 1 (2019).
60. Ritchie, M. E. *et al.* limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic Acids Res.* **43**, e47–e47 (2015).
61. Johnson, W. E., Li, C. & Rabinovic, A. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics* **8**, 118–127 (2007).
62. Leek, J. T., Johnson, W. E., Parker, H. S., Jaffe, A. E. & Storey, J. D. The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics* **28**, 882–883 (2012).
63. Ling, W. *et al.* Batch effects removal for microbiome data via conditional quantile regression. *Nat. Commun.* **13**, 5418 (2022).
64. Warnat-Herresthal, S. *et al.* Scalable prediction of acute myeloid leukemia using high-dimensional machine learning and blood transcriptomics. *Iscience* **23**, 100780 (2020).
65. Pasolli, E., Truong, D. T., Malik, F., Waldron, L. & Segata, N. Machine learning meta-analysis of large metagenomic datasets: tools and biological insights. *PLoS Comput. Biol.* **12**, e1004977 (2016).
66. Kuhn, M. Building predictive models in r using the caret package. *J. Stat. Softw.* **28**, 1–26 (2008).
67. Robin, X. *et al.* proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinf.* **12**, 1–8 (2011).
68. Dhakan, D. et al. The unique composition of indian gut microbiome, gene catalogue, and associated fecal metabolome deciphered using multi-omics approaches. Gigascience **8**, giz004 (2019).
69. Zhang, Y., Patil, P., Johnson, W. E. & Parmigiani, G. Robustifying genomic classifiers to batch effects via ensemble learning. *Bioinformatics* **37**, 1521–1527 (2021).
70. Schirmer, M. *et al.* Dynamics of metatranscription in the inflammatory bowel disease gut microbiome. *Nat. Microbiol.* **3**, 337–346 (2018).
71. Leek, J. T. *et al.* Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat. Rev. Genet.* **11**, 733–739 (2010).

Author contributions

F.S. and Y.L. designed and supervised the study. B.W. implemented the methods, conducted the computational analysis, and drafted the manuscripts. F.S. and Y.L. modified and finalized the manuscripts. All authors read and approved the final version of the manuscript.

Funding

This work was supported by the National Key R &D program of China [grant number 2018YFA0703900] and the National Science Foundation of China [grant number 11971264].

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-024-57670-2.

Correspondence and requests for materials should be addressed to Y.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.