Nucleic Acids Research, 2026, **54**, gkag227
https://doi.org/10.1093/nar/gkag227
Methods

ABaCo: addressing heterogeneity challenges in metagenomic data integration with adversarial generative models

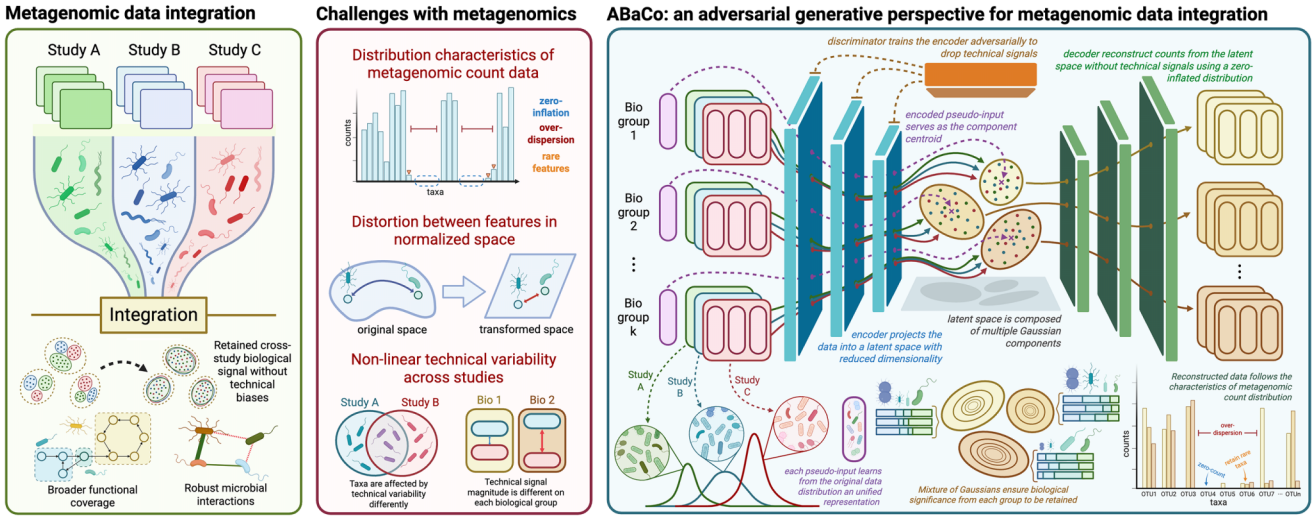
Edir Vidal[1], Angel L. Phanthanourak[1], Atieh Gharib[1], Henry Webel[1], Juliana Assis[1], Sebastián Ayala-Ruano[1], André F. Cunha[1,2], Alberto Santos[1,*]

[1]Novo Nordisk Foundation Center for Biosustainability, Technical University of Denmark, Building 220 Søltofts Plads, Kongens, Lyngby 2800, Denmark
[2]Integrative Biology, Institut Pasteur de São Paulo, São Paulo, SP 05508-020, Brazil
*To whom correspondence should be addressed. Email: albsad@dtu.dk

Abstract

The rapid advancement of high-throughput metagenomics has produced extensive and heterogeneous datasets with significant implications for environmental and human health. Integrating these datasets is crucial for understanding the functional roles of microbiomes and the interactions within microbial communities. However, this integration remains challenging due to technical heterogeneity and the inherent complexity of these biological systems. To address these challenges, we introduce ABaCo, a generative model that combines a variational autoencoder with an adversarial discriminator specifically designed to handle the unique characteristics of metagenomic data. Our results demonstrate that ABaCo effectively integrates metagenomic data from multiple studies, corrects technical heterogeneity, outperforms existing methods, and preserves taxonomic-level biological signals. We have developed ABaCo as an open-source, fully documented Python library to facilitate, support and enhance metagenomics research in the scientific community.

Graphical abstract



Introduction

Advances in sequencing technologies have enabled population-scale metagenomic studies in diverse environments. Several public platforms host extensive collections of microbiome data, enabling a wide range of downstream analyses. For instance, the MGnify platform offers more than 450,000 annotated genomes in Metagenome-Assembled Genomes (MAGs) catalogues, and also provides automated pipelines for the assembly, analysis, and archiving of metagenomic and amplicon sequencing data [1]. Similarly, DOE-JGI's IMG/M offers tools for curation and comparative analysis of microbial genomes and metagenomes [2]. Large collaborative efforts, such as the Human Microbiome Project (HMP), have characterized the healthy human microbiome by analyzing samples from hundreds of individuals [3]. More recently, the Open MetaGenomic dataset (OMG) aggregated over

Received: December 1, 2025. **Revised:** February 13, 2026. **Accepted:** February 21, 2026

3.1 terabases (≈3.3 billion coding sequences) from IMG/M and MGnify [4]. These initiatives demonstrate the scale and diversity of modern microbiome datasets and highlight the potential for atlas-level integrative analyses.

Despite these advances, integrating data from different projects and platforms remains challenging due to technical heterogeneity. Fig. 1a illustrates the potential main sources of variation encountered in the integration of metagenomic datasets. In particular, differences in sequencing technologies (e.g., 16S rRNA amplicon vs shotgun sequencing, or Illumina vs long-read platforms) introduce cross-platform heterogeneity that complicates integration [5]. Additionally, variations in laboratory protocols (DNA extraction, library preparation, etc.) can lead to systematic biases [6]. Even minor protocol variations in metagenomics can substantially alter observed community profiles. These unwanted sources of variation, known as *batch effects*, are unrelated to the biological factors of interest [6]. Batch effects can distort true biological signals, thereby undermining conclusions drawn from cross-study analyses [7]. Correcting for such technical heterogeneity is critical in large-scale metagenomic studies.

Current batch effect correction methods face limitations addressing metagenomic technical heterogeneity (Fig. 1b). Classical methods, originally developed for gene expression data, can be applied to microbiome abundance tables [6]. Parametric linear model approaches such as ComBat [8] and limma [9] employ empirical Bayesian frameworks to adjust for batch effects across features. These methods effectively remove global shifts under the assumption of normality and are widely used in transcriptomics. However, metagenomic data rarely meet these assumptions: read counts are highly sparse and compositional, representing relative abundances that sum to a constant [10]. Ignoring this data structure can lead to misleading inferences [10].

More recent tools designed specifically for metagenomic applications aim to capture the complexity of these datasets better. For example, ConQuR [11] employs a two-part conditional quantile regression model to account for zero-inflation and over-dispersion in microbial read counts, yielding batch-corrected values that preserve the complex count distribution. Another approach, PLSDA-batch [12], uses a multivariate non-parametric method based on partial least squares discriminant analysis. It estimates latent components associated with both biological and technical variation, then subtracts the batch-related components to remove biases while preserving biological differences. However, low-frequency taxa and nonlinear confounding factors can impair these methods' accuracy and complicate downstream interpretation [13].

Large-scale microbiome studies such as metagenomics and metabarcoding demand robust batch correction, yet traditional methods may not fully address the sparse, compositional nature of microbiome counts. While specialized tools incorporate data-specific modeling, several omics fields are increasingly moving towards deep learning solutions that can leverage complex dataset structure with fewer assumptions [13]. Examples from single-cell work, such as ABC [14] and scDREAMER [15] showcase how training an autoencoder with auxiliary cell-type classifiers, along with adversarial alignment, can effectively remove batch effects while preserving biological labels. Most recently, the VampPrior Mixture Model (VMM) [16] has been employed as a prior distribution to the widely used scVI model [17], enhancing its batch correction capability while better retaining biological information in single-cell datasets. These approaches highlight how deep generative modeling can flexibly extract the complexity of data distributions and disentangle technical variation from biological signal.

The methodological parallels in data integration on single-cell transcriptomics and metagenomics are driven by shared challenges: high dimensionality, sparsity of features, and complex confounded technical factors [13]. Here, we introduce **ABaCo**, a generative adversarial framework designed for metagenomic batch correction. ABaCo applies recent advances from single-cell transcriptomics, such as adversarial training and clustering-based priors, and adapts them to the challenges of microbial count data horizontal integration. We demonstrate that ABaCo not only achieves state-of-the-art performance but also removes technical variation across studies while preserving biological signals, potentially improving downstream analyses of microbiome profiles.

Material and methods

ABaCo framework

ABaCo is an adversarial generative framework trained in three sequential phases. In the first phase, a variational autoencoder (VAE) with a variational prior distribution is trained with batch labels supplied to both the encoder and the decoder. This setup also contains two penalization terms–a cross-entropy loss for biological group assignment, and a pairwise KL-divergence between the Gaussian components of the prior distribution–to structure the latent space according to biological groups, thereby promoting biologically meaningful clustering. In the second phase, the prior distribution parameters are set, and adversarial training is applied. The discriminator receives latent embeddings as input and backpropagates a gradient to the encoder. This adversarial setup encourages the encoder to remove batch-specific information from the latent representation while retaining meaningful biological signal. In the third phase, the decoder is trained with batch labels gradually masked. This ensures that the outputs are no longer dependent on batch-specific features. An overview of the model architecture can be found in Fig. 1c, and a more detailed explanation of the theoretical framework in Supplementary File 2.

VAE setup

ABaCo uses the architecture of a regular VAE: an encoder that maps the data into a latent space, a prior distribution that regularizes the latent representation, and a decoder that reconstructs data points back into the original input space [18]. The encoder receives an input data point and outputs the parameters of a Mixture-of-Gaussian (MoG) distribution (means, variances, and mixing probabilities). A latent point is then obtained by first sampling a component from the Categorical distribution using the mixing probabilities, and then sampling from the corresponding Gaussian using the mean and variance. For the prior, ABaCo employs the VampPrior Mixture Model (VMM) as introduced in its original work [16]. This approach learns pseudo-inputs that act as cluster centroids in the latent space, with parameters optimized to represent the data distribution directly in the input space rather than only in the latent space. This provides a more flexible prior that encourages biologically meaningful clustering. The decoder maps the latent representation back to the data space by outputting the parameters of a Zero-inflated Negative Bino-

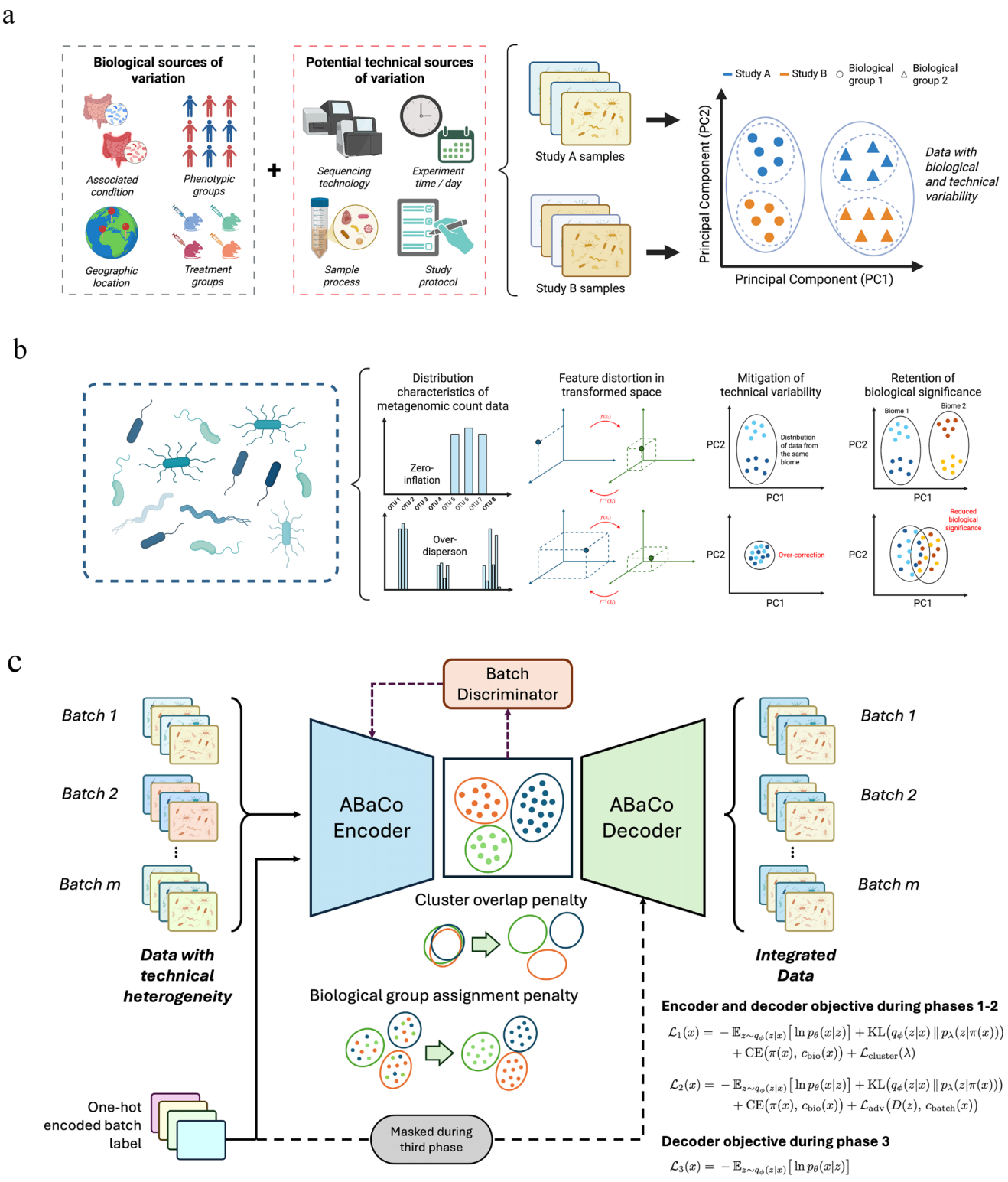


Figure 1 Schematic summary of variation sources, integration challenges and the ABaCo architecture. (**a**) Biological and technical main sources of variation and their effect on sample clustering in metagenomic studies. (**b**) Integration challenges: count distributions (zero-inflation, over-dispersion), transformation distortions, and the trade-off between technical mitigation and biological retention. (**c**) ABaCo architecture, illustrating the VAE, batch discriminator, and the cluster overlap and biological group assignment penalties within a phased training scheme; loss terms are noted alongside the model. Panels (**a**) and (**b**) were created with BioRender.com and are available under a CC BY license at https://BioRender.com/on0jxu5 and https://BioRender.com/52uq2vx, respectively.

mial (ZINB) distribution (mean, dispersion and zero-inflation probability). In this work, the Negative Binomial distribution is also used to compare any meaningful differences in performance.

First phase: Learning parameters of the prior distribution

The VAE training begins with the standard evidence lower bound (ELBO), augmented with additional penalization terms to structure the latent space according to biological groups. The loss function is defined as:

$$\mathcal{L}_1(x) = -\mathbb{E}_{z\sim q_\phi(z|x)}[\ln p_\theta(x|z)] + \mathrm{KL}(q_\phi(z|x) \parallel p_\lambda(z|\pi(x))) + \mathrm{CE}(\pi(x),\, c_{\mathrm{bio}}(x)) + \mathcal{L}_{\mathrm{cluster}}(\lambda) \quad (1)$$

Here, $p_\theta(x|z)$ is the decoder likelihood parameterized by θ, and $q_\phi(z|x)$ is the encoder (approximate posterior) parameterized by ϕ. The variational prior distribution $p_\lambda(z|\pi(x))$ is modeled as a Gaussian mixture with parameters $\lambda = \{\{\mathrm{u_k}\}, \{\sigma_\mathrm{k}^2\}\}_{\mathrm{k}=1}^K$, where $\mathrm{u_k}$ is a pseudo-input that is mapped using the encoder network f_ϕ into the latent space to obtain the component centroid μ_k, and σ_k^2 is the variance of the k-th Gaussian component. The number of mixture components is denoted by K, which corresponds to the number of biological groups being modeled from the data.

The first term, $-\mathbb{E}_{z\sim q_\phi(z|x)}[\ln p_\theta(x|z)]$, is the negative log-likelihood term of the ELBO and works as a reconstruction loss (i.e. how accurately the observed data x can be reconstructed from the latent representation z). The second term, $\mathrm{KL}(q_\phi(z|x) \parallel p_\lambda(z|\pi(x)))$, is the KL-divergence term of the ELBO, which aligns the approximate posterior with the variational prior distribution. Because the prior is a mixture of Gaussians, the component selected for the KL calculation depends on the mixing probabilities $\pi(x)$. The third term, $\mathrm{CE}(\pi(x),\, c_{\mathrm{bio}}(x))$, is a **biological group assignment penalty**, implemented as the categorical cross-entropy between the inferred mixture probabilities $\pi(x)$ and the one-hot encoded biological group $c_{\mathrm{bio}}(x)$. This encourages samples in the same biological group to be mapped to the same latent component. Finally, $\mathcal{L}_{\mathrm{cluster}}(\lambda)$ acts as a **clustering regularizer**, encouraging mixture components to be separated and therefore biological variability between groups to be preserved. Overall, this phase jointly optimizes reconstruction accuracy, prior structure, and biologically meaningful organization of the latent space.

Second phase: Batch correction in the latent space

In the second phase, the prior parameters are set and unaltered for the rest of the training process: learned pseudo-inputs $\mathrm{u_k}$ and variance σ_k^2 of each Gaussian component are fixed. A batch discriminator is then used to correct for any batch effect in the latent space. The discriminator takes latent points z as input and predicts their corresponding batch labels. The loss function for this phase is defined as:

$$\mathcal{L}_2(x) = -\mathbb{E}_{z\sim q_\phi(z|x)}[\ln p_\theta(x|z)] + \mathrm{KL}(q_\phi(z|x) \parallel p_\lambda(z|\pi(x))) + \mathrm{CE}(\pi(x),\, c_{\mathrm{bio}}(x)) + \mathcal{L}_{\mathrm{adv}}(D(z),\, c_{\mathrm{batch}}(x)) \quad (2)$$

Where $D(z)$ is the output of the adversarial network (discriminator), $c_{\mathrm{batch}}(x)$ is the one-hot encoding of the batch label for x, and $\mathcal{L}_{\mathrm{adv}}(D(z),\, c_{\mathrm{batch}}(x))$ is the adversarial loss that backpropagates through the encoder to encourage batch-invariant latent representations. The adversarial loss in this setup is defined as the negative cross-entropy between the discriminator output logits for x and $c_{\mathrm{batch}}(x)$. In this setup, the discriminator learns to correctly predict batch labels, while the encoder is trained to maximize the discriminator's error, effectively removing batch effect from the latent representation (similar to the objective used in the ABC framework [14]). Providing the batch label to the encoder and the decoder biases the model to reconstruct batch-specific features solely from the label; together with the adversarial objective, this encourages a batch-invariant latent z while preserving batch effects in the reconstructed output via the conditional decoder.

Third phase: Decoder batch masking

The third phase starts with the encoder parameters being completely frozen, having only the decoder as a trainable component. The decoder's loss function is limited to the negative log-likelihood:

$$\mathcal{L}_3(x) = -\mathbb{E}_{z\sim q_\phi(z|x)}[\ln p_\theta(x|z)] \quad (3)$$

The idea here is to gradually mask the one-hot encoded batch labels previously provided to the decoder. Eventually, the decoder reconstructs the data solely from the batch-corrected clustered latent space. This procedure forces the decoder to eliminate any residual batch effects while preserving biological signals in the reconstructed data.

Simulated datasets generation

We generated two simulated datasets from a zero-inflated negative binomial (ZINB) distribution. Each scenario – (1) batch effect only, and (2) plus biological effect–consisted of 50 simulated replicates. Each replicate contained 200 samples and 1000 features, with samples assigned to two biological groups in equal proportion (50:50) and two batches in a 60:40 proportion. All draws from the distributions were performed with a fixed seed (42) to ensure reproducibility.

For each feature, the same ZINB parameters were used across all replicates. Specifically, for each feature, the dispersion was sampled as $r \sim \mathrm{Uniform}(1, 3)$ and the baseline log-mean as $\ell_0 \sim \mathcal{N}(2, 1)$; the baseline mean was then obtained by exponentiating ℓ_0. Zero-inflation followed a two-tier scheme: 70% of features were designated as 'highly sparse' with zero-probability $z \sim \mathrm{Uniform}(0.8, 0.9)$, while the remaining 30% of features had $z \sim \mathrm{Uniform}(0.1, 0.8)$.

Batch effect only

For this scenario, the baseline log-mean and variance were modified for samples from Batch 2, while no biological effect was introduced. While batch effects in practice can have broad impacts, we modeled a mixed scenario to evaluate both local and global artifacts. Specifically, for each observation in Batch 2, 10% of the features received an additive shift $\Delta_{batch} \sim \mathcal{N}(1, 2)$ to simulate feature-specific technical biases (a local batch effect). On top of this, batch-wide heteroskedastic adjustments were applied to all features: dispersion shifts $\delta_{disp} \sim \mathcal{N}(0, 0.2)$ and zero-inflation probability shifts $\delta_{zero} \sim \mathcal{N}(0, 0.2)$, representing global, variance-related technical differences.

For a sample in batch $b \in \{0, 1\}$ (Batch $1 = 0$, Batch $2 = 1$), the adjusted mean is:

$$\mu = \exp(\ell_0 + b \cdot \Delta_{batch}) \quad (4)$$

Next, the batch-dependent dispersion is given by:

$$r' = r \cdot (1 + b \cdot \delta_{disp}) \quad (5)$$

Finally, the batch-adjusted zero probability is given by:

$$z' = z \cdot (1 + b \cdot \delta_{zero}) \quad (6)$$

Where $z' \in [0, 1]$ to ensure a valid probability. In order to simulate the ZINB distribution correctly, the counts were sampled from NB(μ, r'), and a zero-mask was applied using Bernoulli(z') to set the corresponding entries to zero.

Batch and biological effect

For the simulated dataset with both batch and biological effects, the procedure extends the batch-only scenario. For 50% of the features in biological group Condition B, an additive log-scale effect was drawn as $\Delta_{bio} \sim \mathcal{N}(2, 2)$. The expected magnitude Δ_{bio} is larger than the batch shift, ensuring that the biological effect is the main source of additive variation. An interaction effect is also included to model biological-specific batch effects on 20% of features. This interaction shift was drawn from $\Delta_{int} \sim \mathcal{N}(3, 1)$ and applied additively on the log-scale mean only for samples belonging to both Batch 2 and Condition B.

For a sample in batch $b \in \{0, 1\}$ (Batch 1 = 0, Batch 2 = 1) with condition $c \in \{0, 1\}$ (Condition A = 0, Condition B = 1), the log-scale mean in:

$$\mu_0 = \exp(\ell_0 + c \cdot \Delta_{bio} + b \cdot \Delta_{batch} + c \cdot b \cdot \Delta_{int}) \quad (7)$$

In addition, a per-sample indicator selects approximately 30% of features. For this subset ($u \in \{0, 1\}$), the mean is adjusted by a feature-wise factor:

$$\mu = \mu_0 \cdot \left(1 + u \cdot \frac{\Delta_{bio}}{2 \cdot \max(\Delta_{bio})}\right) \quad (8)$$

Dispersion and zero-inflation probabilities are obtained in the same way as in the batch-only simulations, with the counts drawn from the resulting ZINB distribution.

Case study datasets

Anaerobic digestion dataset

The anaerobic digestion dataset was used in a study benchmarking several batch effect assessment and correction approaches in microbiome data [6]. The data is publicly available through the GitHub repository linked to the work, where it was downloaded and converted to a comma-separated values (CSV) file for convenience. As mentioned before, it consists of 567 taxonomic groups and 75 samples, distributed across 5 batches (processing days) and 2 biological groups (treatment groups). The implementation is available as a demonstration in a Jupyter Notebook within the ABaCo repository.

Inflammatory bowel disease dataset

The dataset is composed of two different BioProjects which were merged directly before pre-processing: one that studies the dynamics of microbiome functionality in IBD (accession: PRJNA389280) and another that includes multi-omics measurements from IBD patients in a longitudinal study (accession: PRJNA398089). Both studies are also available in MGnify (study IDs MGYS00002301 and MGYS00006120 respectively), from which the overall taxonomic assignments were downloaded. For consistency, only the genus-level taxa were retained, resulting in a dataset comprising 193 taxonomic groups and 517 samples. Metadata was retrieved from the Data Repository for Human Gut Microbiota (GMrepo) [19], where the phenotype identifiers were mapped to categorical labels (i.e. D003425 →Crohn Disease, D003093 →Ulcerative Colitis, and D006262 →non IBD).

DTU-GE sewage dataset

For the DTU-GE sewage dataset, we retrieved the taxonomic abundance tables directly from the MGnify API for the analysis accessions listed in the study MGYS00001312. Data were obtained separately using the SSU rRNA taxonomic analysis from pipeline versions 4.1 and pipeline 3.0, capturing batch effect associated with computational framework differences. Counts were aggregated to the phylum level, and annotated with the geographic location of sampling to be used as a desirable co-founding variable (biological group). We retained samples from countries represented at least 15 times in the whole dataset, resulting in a total of 4 countries and 129 samples. Finally, taxa with zero counts across all samples were removed, yielding a final dataset of 162 taxonomic features.

Performance assessment

We used four metrics generally used to quantify batch-effect correction in single-cell studies [20]: kBET [21], iLISI [22], batch ASW [23] and batch ARI [24] - to assess batch effect removal, where values closer to 1 indicate better performance. All metric results for each method across all datasets, as well as ABaCo per-run results, are provided in Supplementary File 3.

We applied PCoA using the Aitchison distance to compare clustering by batch and biological group, both before and after correction. To quantify the variance explained by batch and biological factors, we performed a Permutational Multivariate Analysis of Variance (PERMANOVA) test with 999 permutations, reporting the R^2 attributable to each factor independently [25].

For the robustness analysis, we track the 5 most abundant taxa in each case study dataset after correction using ABaCo by reporting their mean relative abundance across all runs. We report the mean and variance for each biological group to verify the consistency of ABaCo's inference in each iteration. Additionally, we applied the Kruskal-Wallis test [25] to all taxonomic features to identify significant differences among biological groups before and after correction. For each taxon, we computed the K-W statistic and its p-value, adjusting for multiple testing across taxa using the Benjamini-Hochberg false discovery rate. Taxa with an adjusted P-value below 0.05 were considered significantly differentially abundant in at least one biological group (see Supplementary File 1, Supplementary Fig. S12).

State-of-the-art methods for batch effect correction use either normalized counts (Batch Mean Centering (BMC) [6], ComBat [8], limma [9], PLSDA-batch [12]) or raw counts (ComBat-seq [26], ConQuR [11] and ABaCo (ours)). Because ABaCo implements a generative model, all analyses were repeated for 50 training runs, and we report both mean performance and variability to quantify the consistency of the model.

Experimental setup

The model architecture and training hyperparameters were kept consistent across all datasets. The encoder consists of 512, 256, and 128 neurons, respectively, and the decoder mirrors this with three layers of 128, 256, and 512 neurons. The latent space dimension was set to 16 for all datasets except the IBD case study, where 32 dimensions were used. ReLU ac-

tivations were applied between each layer. The batch discriminator network consists of three fully connected layers (256, 128, and 64 neurons) with ReLU activations. A summary of the training hyperparameters used in each dataset is provided in the Supplementary File 1, Supplementary Table S1. The epochs for each phase (1st, 2nd, and 3rd) and the values of all learning rates used—Phases lr.: VAE learning rates for each phase; Disc. lr.: learning rate used when training the discriminator during the second phase; Adv. lr.: learning rate used to backpropagate the adversarial loss to the encoder—is defined in here. The weights for each term on the losses are also mentioned in the Supplementary File 1, Supplementary Table S2. NLL: negative log-likelihood term of the ELBO; KL-div: KL-divergence between prior $p_\lambda(z)$ and posterior $q_\phi(z|x)$ distributions; Bio. penalty: cross-entropy used for the biological group assignment penalty; Clust. penalty: pairwise KL-divergence of prior Gaussian components used for the cluster overlap penalty.

Model training and evaluation were performed on a dual-socket Intel Xeon Gold 6226R server running in x86_64 mode with VT-x virtualization enabled. Each socket contains 16 physical cores with Hyper-Threading (2 threads per core), yielding a total of 64 logical CPUs. The system is equipped with 1 TiB of RAM and runs Debian 12 'Bookworm' with Linux kernel 6.1.0-37-amd64. For GPU acceleration, two NVIDIA Tesla V100S-PCIE cards (32 GiB each) were installed, using NVIDIA driver 565.57.01 and CUDA 12.7. Despite relying on an adversarial generative framework, the method is notably efficient with a running time of 3 minutes or under on every dataset (see Supplementary File 1, Supplementary Table S3 for the summarized compute performance).

Computational performance

In order to evaluate ABaCo's computational performance and requirements, we used Python's libraries GPUtil, psutil and time to calculate wall-clock running times and peak GPU memory usage. The calculations have been included in the notebook report-performance.ipynb, available in ABaCo's documentation (https://mona-abaco.readthedocs.io/en/latest/tutorial/report-performance.html) (see Supplementary File 1, Supplementary Fig. S17).

ABaCo Python library

All the code for ABaCo is openly available on GitHub (https://github.com/Multiomics-Analytics-Group/abaco) under the MIT license, and is distributed as a Python package in PyPI (https://pypi.org/ project/abaco/). The documentation is provided at ReadtheDocs (https://mona-abaco.readthedocs.io/), where there are several tutorials exemplifying how to use ABaCo.

Results

Datasets overview

We benchmarked ABaCo (Fig. 1c) against state-of-the-art methods across scenarios to evaluate performance and robustness. We created simulated count datasets to establish a controlled environment with known ground-truth biological groups (simulated datasets). For these datasets, all benchmarking and statistical results are computed over 50 independent replicates per scenario. Additionally, we assessed ABaCo's performance on horizontal data integration tasks using a series of independent datasets from different sources (case studies), including anaerobic digestion assays, inflammatory bowel disease (IBD) patient samples, and sewage metagenomic profiles. As shown in Fig. 2a, both the simulated datasets and the case studies exhibit comparable distributions with a highly skewed proportion of zero counts, indicating widespread zero inflation. This is an inherent characteristic of metagenomic data, where a large subset of taxonomic features is absent in most samples, resulting in a long left tail in the distribution of non-zero proportions.

Simulated datasets

We generated two complementary sets of simulated datasets sampling from the zero-inflated negative binomial (ZINB) distribution to evaluate model performance. The first scenario included only batch effects with no biological differences between groups, while the second scenario contained both biological signal and batch differences. Each replicate consisted of 200 samples and 1,000 features, divided into two biological groups (50:50) and two batches (60:40). To focus evaluations on sampling variability, the parameters of the sampling distribution were kept constant across replicates. We created 50 independent replicates for each scenario and used the resulting counts to evaluate the model's batch correction performance and consistency.

Fig. 2b illustrates the distributions of Shannon and Simpson alpha diversity indices stratified by batch and biological groups, along with the beta diversity Aitchison distances Principal Coordinate Analysis (PCoA) for the simulated data with batch and biological effects. Both alpha diversity indices display a clear shift concordant with the imposed batch and biological effects: the two biological groups differ in median alpha diversity and in the spread of values, as do the two batch groups. Additionally, the PCoA reveals strong partitioning on both batch and biological effects, with the biological effect being the primary driver. This is confirmed by the PERMANOVA R^2 value, which quantifies the variance explained by batch and biological factors separately. These results indicate that the simulated biological signal is the primary source of compositional differences, while the simulated batch effect contributes a smaller, yet visually evident, source of variation. The same illustration can be found on Supplementary File 1, Supplementary Fig. S1 for the simulated dataset with batch effect only.

Case studies datasets

To evaluate ABaCo's performance across diverse case studies, we selected datasets with a wide range of characteristics. An overview of these is provided in Table 1, detailing the batch and biological groups. Notably, the anaerobic digestion is a benchmark dataset used for microbiome batch effect assessment [6]. The integrated IBD studies contained highly unbalanced batches and heterogeneous disease groups. Finally, the DTU-GE sewage datasets are semi-balanced across batches that correspond to different preprocessing pipeline versions, with nearly equal sample sizes per batch. Both the DTU-GE sewage and IBD datasets show high variability in the relative abundance of the most dominant taxa. Importantly, the separation between biological groups is not consistent across batches, indicating that batch-specific effects may influence the biological inference. To assess the consistency of ABaCo,

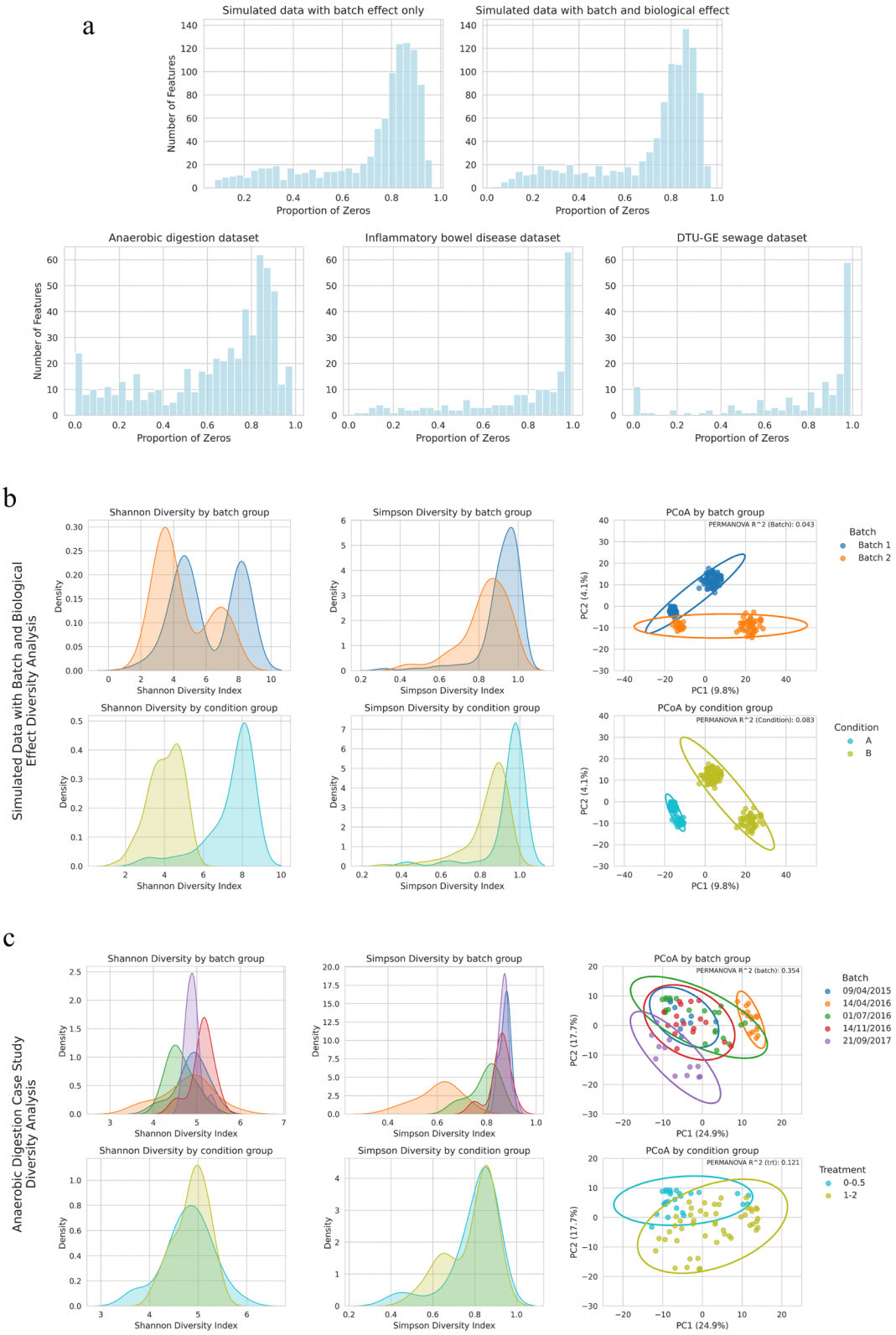


Figure 2 Comprehensive comparison of data sparsity, alpha diversity and beta diversity across simulated and case study datasets. (**a**) Histograms of the proportion of zero counts per taxonomic feature for simulated and case study datasets. (**b** and **c**) Kernel density estimates of Shannon and Simpson alpha diversity (left, center) and PCoA on Aitchison distances (right), shown by batch (top), and biological group (bottom) for the simulated dataset (b) and the anaerobic digestion case study (c).

Table 1. Overview of datasets, batches, biological groups, and sample counts for every case study. For each dataset, the table reports the number of taxa, the batch group identifier (date, accession or pipeline), the biological-group labels present in that batch (treatment, phenotype or location) and the per-group sample counts; the right-most column gives the total number of samples in each batch and dataset

Dataset	Taxa	Batch	Biological group	Samples	Batch total
Anaerobic digestion	567	09/04/2015	0–0.5	4	9
			1.0–2.0	5	
		14/04/2016	0–0.5	4	16
			1.0–2.0	12	
		01/07/2016	0–0.5	8	21
			1.0–2.0	13	
		14/11/2016	0–0.5	8	17
			1.0–2.0	9	
		21/09/2017	0–0.5	2	12
			1.0–2.0	10	
Inflammatory Bowel Disease	193	PRJNA389280	Crohn's Disease (CD)	175	341
			Ulcerative Colitis (UC)	97	
			non IBD	69	
		PRJNA398089	Crohn's Disease (CD)	85	176
			Ulcerative Colitis (UC)	46	
			non IBD	45	
DTU-GE sewage	162	pipeline 4.1	Canada	11	56
			USA	30	
			Zambia	7	
			Australia	8	
		pipeline 3.0	Canada	13	73
			USA	40	
			Zambia	10	
			Australia	10	

we trained the model 50 times on each dataset with different initialization, using a defined seed (42) only for the first training instance.

Anaerobic digestion study: this benchmark dataset, used for microbiome batch effect assessment [6], comprises anaerobic digestion of organic matter under varying phenol concentrations [27]. The dataset includes 567 Operational Taxonomic Units (OTUs) identified from the microbiota of 75 samples collected in batches on different dates, which is the main source of technical variation. Fig. 2c provides an overview of the alpha and beta diversity of the dataset. The distribution of Alpha diversity indices indicates that batch effects are the primary source of variation, as evidenced by lower overlap between batches compared to biological groupings. The PCoA further confirms this effect, showing that the biological effect remains significantly different between the two treatment groups despite the batch variation.

Inflammatory bowel disease projects: this case study comprises two different studies [28] [29] from Harvard T.H. Chan School of Public Health, available in MGnify, which analyze the microbiome dynamics of 517 patients with inflammatory bowel disease (IBD) and non-IBD controls. Within the IBD and non-IBD patient groups, a total of 193 genus-level taxa were identified, with cross-study batch effect accounting for the main source of variation in the dataset (Supplementary File 1, Supplementary Fig. S2). A distinction was made between Crohn's disease (CD) and Ulcerative colitis (UC) for a more enriched analysis of the phenotypes' taxonomic differences.

DTU-GE Sewage global surveillance: this study involves global sewage metagenome surveillance for pathogen and antimicrobial-resistance monitoring from the National Food Institute, Technical University of Denmark (DTU-GE) [30]. It includes point–prevalence metagenomic profiles from raw sewage collected at main sewer inlets of major cities worldwide. The data were preprocessed using two different MGnify pipeline versions at the phylum level, which introduces technical heterogeneity. For our analysis, we focused on samples from Australia, the USA, Canada, and Zambia, as each of these countries contributed at least 10 samples. After filtering, 162 taxa were retained with each pipeline accounting for the main source of variation in the data (Supplementary File 1, Supplementary Fig. S3).

ABaCo outperforms state-of-the-art batch correction methods

For a thorough evaluation of our model, it is essential to acknowledge the challenges posed by the complexity and heterogeneity of batch effects in biological datasets [31]. To address these issues effectively, we adopted a holistic approach that evaluates both the efficacy of batch effect correction and the preservation of biological signals, reporting graphical representations and quantitative metrics.

ABaCo was trained multiple times with different initializations for the case studies and with a defined seed (42) for the simulated datasets to test robustness. The training hyperparameters and weights used for each scenario are detailed in Supplementary File 1, Supplementary Tables S1–S2. ABaCo was tested using two output distributions to assess whether actively modeling the sparsity of the data would substantially improve the performance. More specifically, we tested the zero-inflated negative binomial (ZINB) and negative binomial (NB) distributions using the same model hyperparameters.

We also applied state-of-the-art methods to correct the datasets using normalized counts (BMC, ComBat, limma and PLSDA-batch) and raw counts (ComBat-seq and ConQuR) to compare the performance of ABaCo. Fig. 3 summarizes the batch correction performance using various metrics for all the mentioned methods, where values closer to 1.0 reflect improved batch correction performance. To assess local

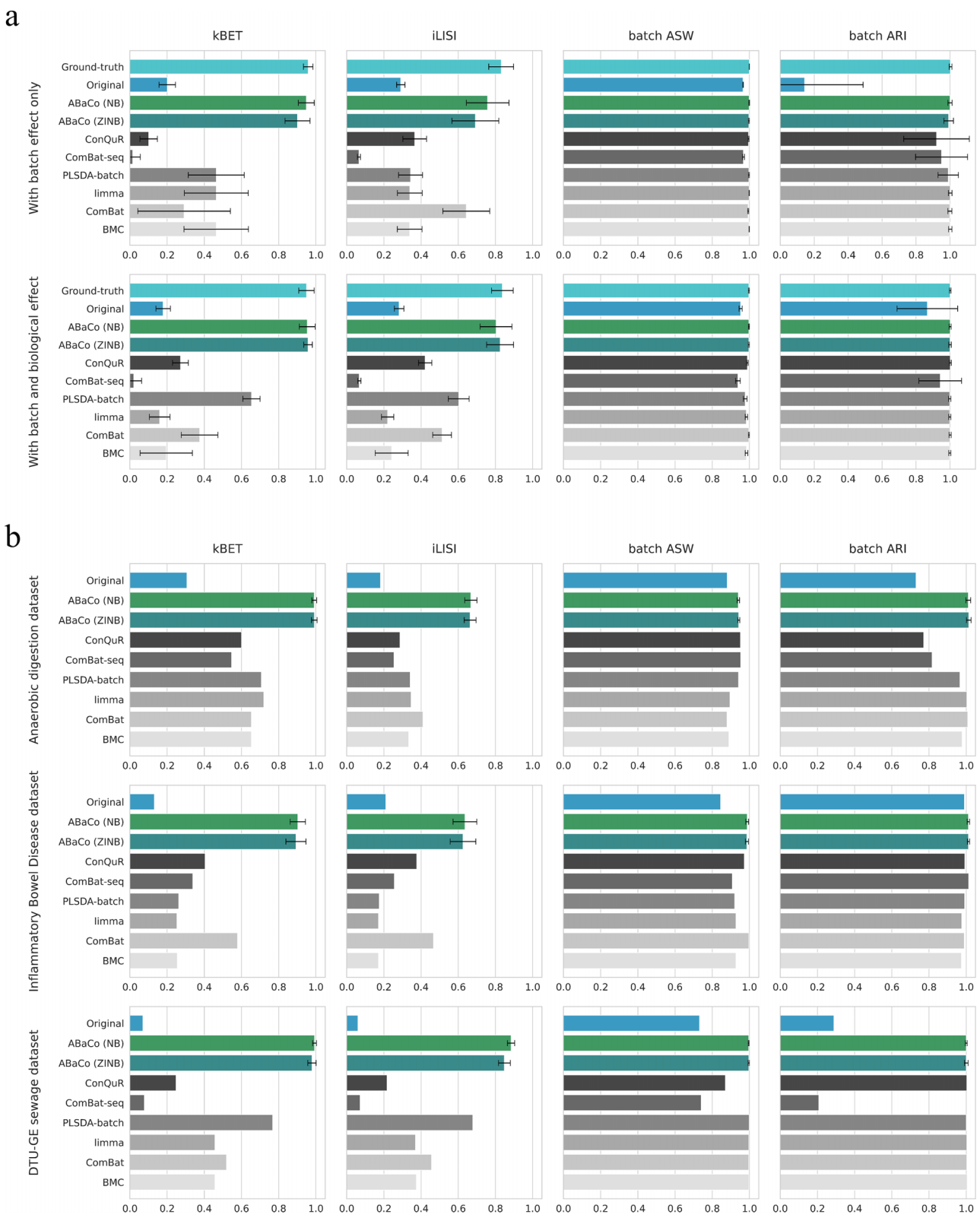


Figure 3 Performance of ABaCo and state-of-the-art methods on simulated and case study datasets. (**a**) In simulated datasets where the ground-truth is reported, ABaCo performs well both with and without biological effects, outperforming alternatives in local batch correction (kBET, iLISI) while maintaining global structure comparably (batch ASW and ARI). (**b**) In case studies, ABaCo likewise surpasses state-of-the-art methods (kBET, iLISI) and preserves global structure to a similar extent (batch ASW and ARI).

batch mixing and sample-level integration, we employed the k-nearest neighbor Batch Effect Test (kBET) [21] and the integrated Local Inverse Simpson's Index (iLISI) [22]. In addition to these, we evaluated global structure and cluster overlap using batch Average Silhouette Width (ASW) [23] and batch Adjusted Rand Index (ARI) [24].

In the simulated datasets, ABaCo with both negative binomial (NB) and zero-inflated negative binomial (ZINB) distributions achieves the best performance in batch effect correction compared to other methods for the kBET and iLISI metrics (Fig. 3a). For the batch ASW and batch ARI metrics, ABaCo performs similarly to the top methods. Notably, ABaCo with the ZINB distribution showed lower variance and higher mean performance in the kBET and iLISI metrics for the simulated data with both batch and biological effects. In the case studies, ABaCo outperforms all the other methods in the kBET and iLISI metrics while preserving the biological signal, as indicated by comparable batch ASW and ARI metrics (Fig. 3b).

In Fig. 4a, the PCoA showcases that ABaCo effectively integrates the batch groups of the simulated data containing both batch and biological effects. With the ZINB model, ABaCo achieves the closest alignment to the ground-truth PCoA plot, successfully mixing the batches and preserving the biological structure. ABaCo also handles batch effects without adding biological bias for the simulated data with batch effect only (Supplementary File 1, Supplementary Fig. S4).

The benchmarking metrics demonstrate that ABaCo effectively addresses batch effects from multiple groups. This is also visible and confirmed in the figures for the anaerobic digestion dataset (Fig. 4b) and the other two case studies (Supplementary File 1, Supplementary Figs S5–S6), where PCoA plots show the batch effects corrected and the biological differences maintained. We did the same analysis benchmarking against state-of-the-art methods for all datasets (Supplementary File 1, Supplementary Figs S7–S11).

ABaCo provides consistent results across multiple runs

We assessed ABaCo's robustness by tracking the most abundant taxa in each case study after batch correction across all training runs. Fig. 5 reports the mean relative abundance of the five most abundant taxa, stratified by biological group. Statistical comparisons between groups were performed using the Kruskal–Wallis test, and the relative abundances of the two most abundant taxa were plotted in detail (Supplementary File 1, Supplementary Figs S12–S13).

For the anaerobic digestion case study, both ZINB and NB models return very low variance across runs for the top five taxa (variance range, indicating consistent reconstruction of dominant taxa). In the IBD and DTU-GE datasets, the most abundant taxa were likewise preserved: the two most frequent taxa together account for more than 50% of total counts in their respective datasets. Variance for the two most abundant taxa was higher than for lower-ranked taxa, which is a reflection of the heterogeneity within groups already visible in the original data (Supplementary File 1, Supplementary Figs. S13–S16). Overall, these results show that ABaCo reliably preserves the dominant taxonomic features across runs, with larger variance concentrated in the most abundant taxa.

Discussion

We have presented ABaCo, a generative adversarial framework for correcting technical heterogeneity while preserving biological signals in horizontal metagenomic integration. ABaCo combines a VAE with an adversarial discrimination approach to correct count-based distributions (NB) and accounting for zero-inflation (ZINB). Across both simulated and real-world case studies, ABaCo consistently outperformed state-of-the-art methods, reducing batch-associated structure, as evidenced by high-scoring kBET and iLISI metrics, and preserving biological separation, as demonstrated by biologically informed PCoA plots and test statistics. Additional examples of use cases are available in ABaCo's documentation, showing that this performance is maintained when integrating a high number of diverse studies (integration of nine studies - https://mona-abaco.readthedocs.io/en/latest/ tutorial/demo-parkinson.html).

Explicitly modeling count distributions proved beneficial for performance on sparse microbiome data, with the ZINB model demonstrating lower variance in challenging simulated settings, aligning well with observed zero-inflation patterns [32]. Additionally, the adversarial discriminator effectively removed provenance signals in the latent space, highlighting its strengths in mitigating batch effects, crucial given that technical variance can dominate sequencing-derived profiles even in tightly controlled experiments [33]. The balance between adversarial and biological-preservation losses is crucial, as it influences the trade-off between batch removal and the conservation of biological signals. To optimize this trade-off, we recommend using a small validation set when applying ABaCo to new datasets. This approach allows for fine-tuning the weight of the adversarial loss, ensuring robust batch effect removal while preserving biologically meaningful variations (see https://mona-abaco.readthedocs.io/en/latest/tutorial/report-performance.html).

ABaCo effectively models observed counts and zero inflation. While it does not intrinsically enforce compositional constraints, thoughtful preprocessing choices such as filtering rare taxa, library-size normalization, or compositional transforms can improve downstream analyses and should be consistently reported to ensure reproducibility. In addition, adversarial training can be sensitive to optimizer schedules and discriminator–encoder balance. By implementing carefully staged learning rates, we reduced mode collapse, demonstrating ABaCo's adaptability to large datasets. Furthermore, the latent clustering prior helped preserve biological group structure in our experiments, indicating its potential for maintaining meaningful biological signals. Future work could focus on improving the interpretability of latent features, linking them to environmental covariates or phylogenetic structures to gain deeper biological insights.

Robustness analyses, which assess consistency across runs or variance in output behavior, are still relatively uncommon for generative models. However, they are crucial for evaluating model reliability beyond best-case scenarios, diagnosing sources of instability, and setting realistic expectations for deployment [34]. To this end, we conducted such analyses and found that, on average, run-to-run reproducibility of the most abundant taxa after batch correction was high across datasets, indicating highly consistent reconstruction of dominant community members. These results support the claim that ABaCo reduces technical heterogeneity without erasing biologically

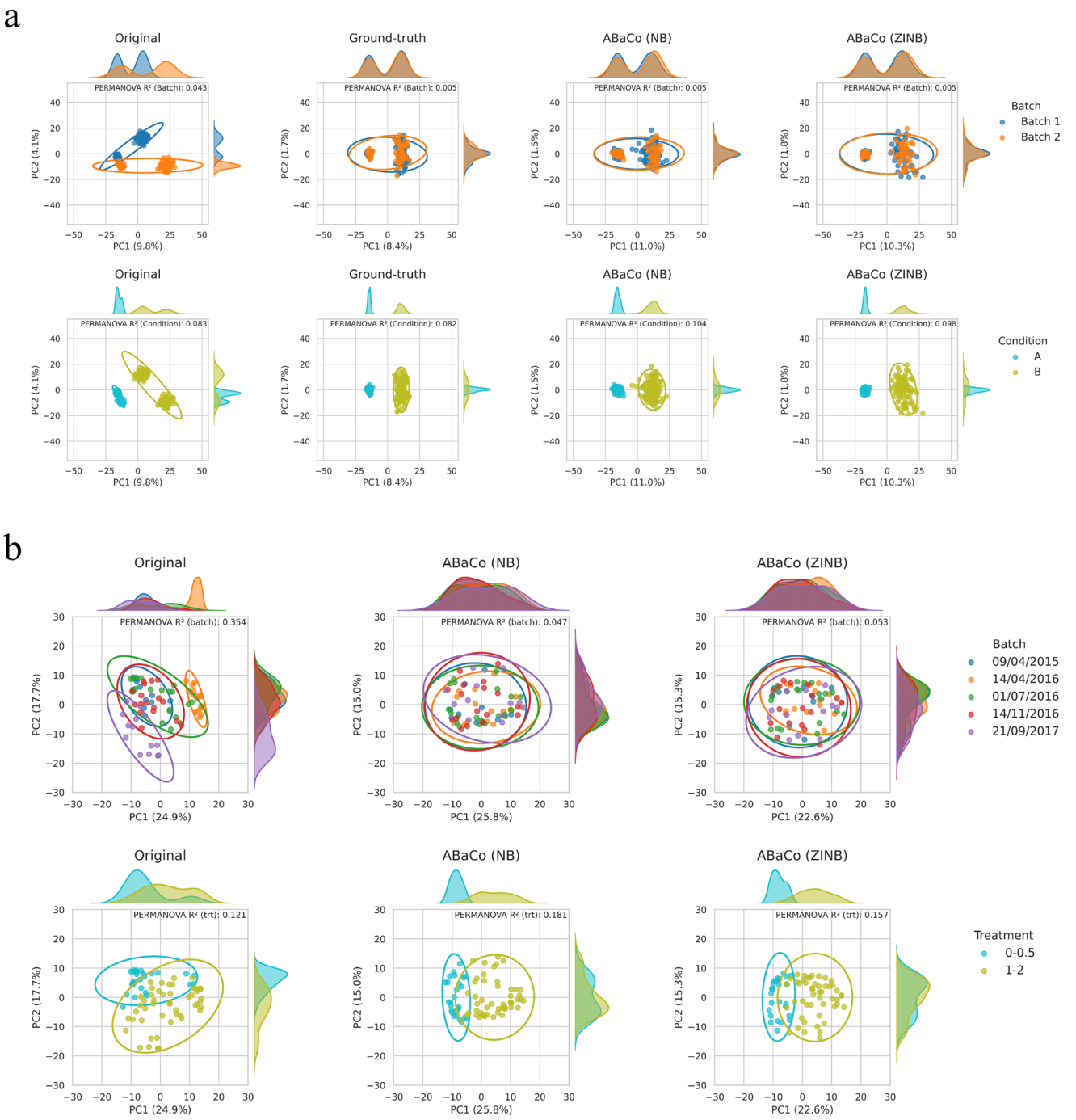


Figure 4 Principal Coordinates Analysis (PCoA) using Aitchison distance on datasets corrected with ABaCo (ZINB and NB models). (**a**) Simulated data containing both batch and biological effects; the ground-truth PCoA is shown for reference. (**b**) Anaerobic digestion case study. For each dataset, ABaCo mix batch groups while preserving biologically relevant group structure.

meaningful variation. This pattern also suggests that larger absolute abundances amplify run-to-run variance even when overall taxonomic structure is preserved, and that highly heterogeneous cohorts (e.g. IBD) can produce a small number of extreme outlier runs. Human samples, and particularly IBD samples, may be more variable across individuals and studies, and the combination of high biological heterogeneity across batches may amplify sensitivity to initializations or hyperparameters [35]. Therefore, we recommend that analyses of similarly heterogeneous atlases include multiple training replicates, inspection of low-agreement runs, and reporting of both mean tendency and tail statistics rather than relying solely on averages (see https://mona-abaco.readthedocs.io/en/latest/tutorial/tutorial-ibd.html).

Beyond ABaCo's capabilities for understanding and reconstructing a community structure, one of the main goals is to facilitate downstream comparative analyses, such as differential abundance testing. We acknowledge that any batch correction procedure inherently modifies the data distribution, which can have implications during statistical hypothesis testing. However, ABaCo's generative framework is explicitly designed to preserve the native probabilistic structure of the microbiome,

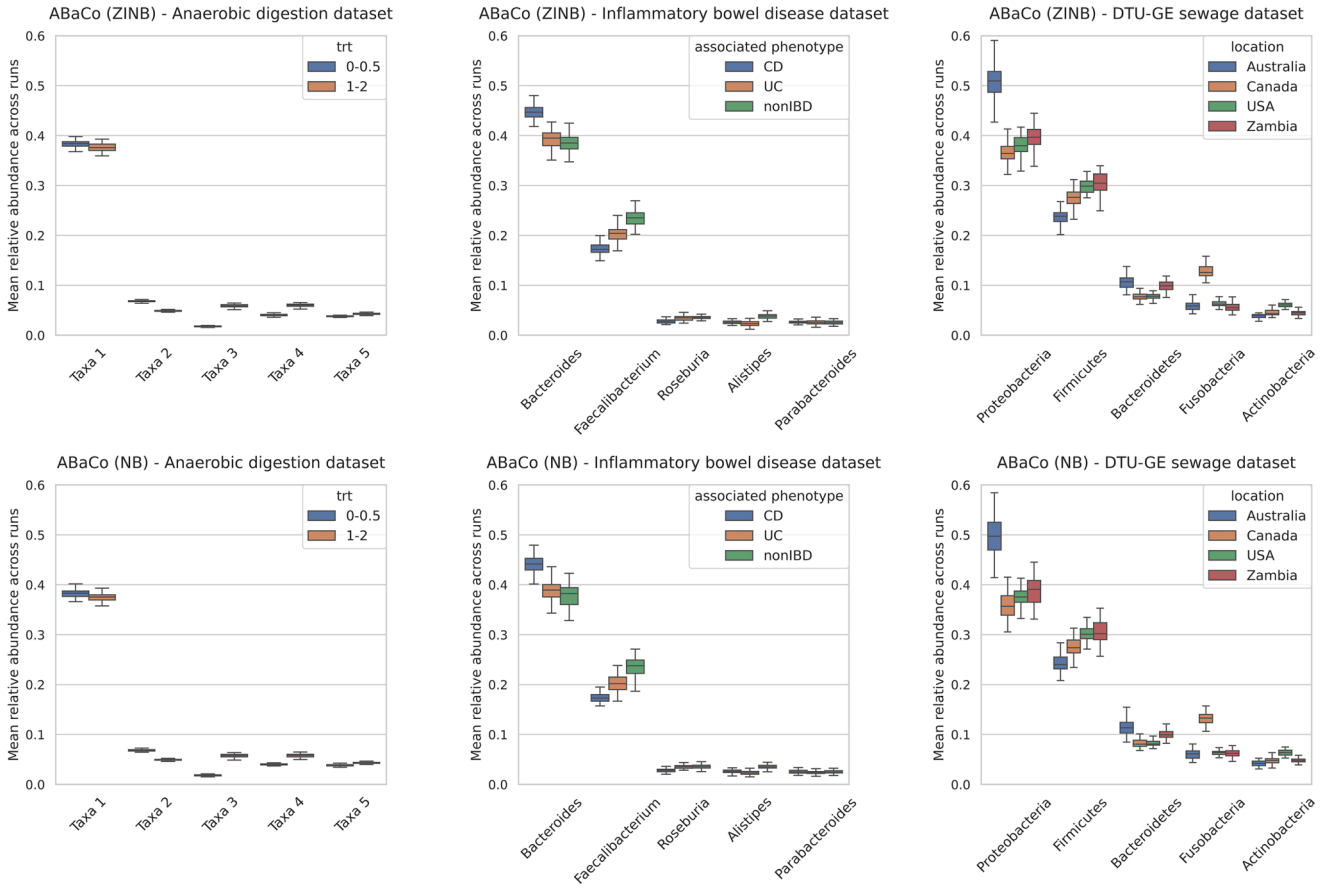


Figure 5 Box-plots of the five most abundant taxa mean (relative abundance) for all training iterations of ABaCo (Anaerobic digestion dataset does not provide taxonomic resolution). Mean value is registered after every training iteration with the grouping based on condition (treatment group, associated phenotype, or geographic location).

providing profiles that are not only integrated without technical signals but also with their biological signatures preserved. In this sense, we regard the corrected outputs from the model as suitable for downstream inference, though it's important to keep in mind that reconstruction variance is naturally higher in low-signal taxa.

While ABaCo demonstrates robust performance, it is important to pinpoint the applicability limitations to scenarios with strong confounding or highly unbalanced sampling where biological and technical sources are deeply entangled. One good example of such boundaries involves cross-platform heterogeneity (e.g. 16S rRNA amplicon vs. shotgun sequencing); as our current benchmarking has only been focused on shotgun metagenomics, the framework's efficacy in broader cross-modality integration remains to be evaluated. To further enhance ABaCo's capabilities, future developments could incorporate explicit compositional models to boost biological fidelity [36]. Prior extensions could increase latent expressivity and better decouple batch effects from biological signals [18]. For instance, addressing the inconsistency of the latent space, which is common on VAEs, could improve robustness in highly heterogeneous cohorts [37]. Expanding the evaluation to include targeted downstream analyses presents a natural next step to demonstrate the practical benefits of ABaCo's batch correction. Incorporating analyses such as differential abundance with FDR control, network inference, and functional profiling would help show how batch correction influences biological conclusions [38]. We note that some of these evaluations are already possible with our outputs, but a full biological interpretation of the results is beyond the scope of the current paper. Nonetheless, we anticipate that future studies leveraging ABaCo should reveal enhanced biological relevance and actionable insights, paving the way for a more reliable and impactful microbiome research.

To ensure wide adoption, accessibility, and reproducibility, we developed ABaCo as an open-source Python library. It is readily available via PyPI for easy installation and integration into custom workflows. Comprehensive documentation enables rapid implementation of batch correction and allows the community to build upon our work and contribute to further develop ABaCo. Additionally, we also provide benchmark simulated datasets and associated code to quantitatively assess the performance of future batch effect correction methods. By establishing this open and consistent benchmarking resource, we aim to accelerate innovation and elevate the quality of research across the entire field.

Acknowledgements

We thank Anders Krogh for advice on the model design and the Novo Nordisk Foundation Center for Biosustainability Informatics Platform for technical support during the implementation of ABaCo. The authors acknowledge that the graphical abstract was created using BioRender.com and is used under a

CC BY publication license available athttps://BioRender.com/8yhwosp, as well as panels (a) and (b) from Fig. 1 (links available in Fig. 1 caption).

Author contributions: A.S. conceived and supervised the project. E.V. led the design of the ABaCo pipeline with theoretical guidance from A.G. and H.W. E.V., A.L.P., and S.A. implemented and maintained the code repository and documentation. A.L.P. set up the ABaCo Python library and created several tutorials demonstrating its use. E.V., A.G., and A.S. wrote the paper with critical revision from H.W., J.A., S.A., and A.F.C. All authors reviewed and approved the final manuscript.

Supplementary data

Supplementary data is available at NAR online.

Conflict of interest

None declared.

Funding

This work was supported by the Novo Nordisk Foundation [NNF20CC0035580] and Calmette & Yersin PhD Grant (Pasteur Network, Sep/2023). Funding to pay the Open Access publication charges for this article was provided by the Novo Nordisk Foundation [NNF20CC0035580].

Data availability

The code and data underlying this article are available in Zenodo, at https://doi.org/10.5281/zenodo.18483019. The datasets were derived from sources in the public domain: MGnify - EBI at https://www.ebi.ac.uk/metagenomics. All the code for ABaCo is openly available on GitHub (https://github.com/Multiomics-Analytics-Group/abaco) under the MIT license, and is distributed as a Python package in PyPI (https://pypi.org/project/abaco/). The documentation is provided at ReadtheDocs (https://mona-abaco.readthedocs.io/) where there are several tutorials exemplifying how to use ABaCo.

References

1. Richardson L, Allen B, Baldi G *et al.* MGnify: the microbiome sequence data analysis resource in 2023. *Nucleic Acids Res* 2023;51:D753–9. https://doi.org/10.1093/nar/gkac1080
2. Chen IMA, Chu K, Palaniappan K *et al.* The IMG/M data management and analysis system v.7: content updates and new features. *Nucleic Acids Res* 2023;51:D723–32. https://doi.org/10.1093/nar/gkac976
3. Turnbaugh PJ, Ley RE, Hamady M *et al.* The human microbiome project. *Nature* 2007;449:804–10. https://doi.org/10.1038/nature06244
4. Cornman A, West-Roberts J, Camargo AP *et al.* The OMG dataset: An Open MetaGenomic corpus for mixed-modality genomic language modeling. In: The Thirteenth International Conference on Learning Representations. *Biorxiv.* 2025. https://doi.org/10.1101/2024.08.14.607850
5. Durazzi F, Sala C, Castellani G *et al.* Comparison between 16S rRNA and shotgun sequencing data for the taxonomic characterization of the gut microbiota. *Sci Rep* 2021;11:3030. https://doi.org/10.1038/s41598-021-82726-y
6. Wang Y, LêCao KA. Managing batch effects in microbiome data. *Brief Bioinform* 2020;21:1954–70. https://doi.org/10.1093/bib/bbz105
7. Fachrul M, Méric G, Inouye M *et al.* Assessing and removing the effect of unwanted technical variations in microbiome data. *Sci Rep* 2022;12:22236.https://doi.org/10.1038/s41598-022-26141-x
8. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;8:118–27. https://doi.org/10.1093/biostatistics/kxj037
9. Smyth GK. limma: linear models for microarray data. In: *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*. New York: Springer-Verlag, 2005, 397–420. https://doi.org/10.1007/0-387-29362-0_23
10. Gloor GB, Macklaim JM, Pawlowsky-Glahn V *et al.* Microbiome datasets are compositional: and this is not optional. *Front Microbiol* 2017;8: 2224.https://doi.org/10.3389/fmicb.2017.02224
11. Ling W, Lu J, Zhao N *et al.* Batch effects removal for microbiome data via conditional quantile regression. *Nat Commun* 2022;13:5418. https://doi.org/10.1038/s41467-022-33071-9
12. Wang Y, Lê Cao KA. PLSDA-batch: a multivariate framework to correct for batch effects in microbiome data. *Brief Bioinform* 2023;24:1–17.
13. Yu Y, Mai Y, Zheng Y *et al.* Assessing and mitigating batch effects in large-scale omics studies. *Genome Biol* 2024;25:254. https://doi.org/10.1186/s13059-024-03401-9
14. Danino R, Nachman I, Sharan R. Batch correction of single-cell sequencing data via an autoencoder architecture. *Bioinform Adv* 2023;4:vbad186. https://doi.org/10.1093/bioadv/vbad186
15. Shree A, Pavan MK, Zafar H. scDREAMER for atlas-level integration of single-cell datasets using deep generative model paired with adversarial classifier. *Nat Commun* 2023;14:7781. https://doi.org/10.1038/s41467-023-43590-8
16. Stirn AA, Knowles DA. The VampPrior Mixture Model. arXiv, https://doi.org/10.48550/arXiv.2402.04412, 11 March 2025, preprint: not peer reviewed.
17. Lopez R, Regier J, Cole MB *et al.* Deep generative modeling for single-cell transcriptomics. *Nat Methods* 2018;15:1053–8. https://doi.org/10.1038/s41592-018-0229-2
18. Tomczak JM. Deep Generative Modeling. Cham, Switzerland: Springer Nature, 2024. https://doi.org/10.1007/978-3-031-64087-2
19. Dai D, Zhu J, Sun C *et al.* GMrepo v2: a curated human gut microbiome database with special focus on disease markers and cross-dataset comparison. *Nucleic Acids Res* 2021;50:D777–84. https://doi.org/10.1093/nar/gkab1019
20. Tran HTN, Ang KS, Chevrier M *et al.* A benchmark of batch-effect correction methods for single-cell RNA sequencing data. *Genome Biol* 2020;21:12. https://doi.org/10.1186/s13059-019-1850-9
21. Büttner M, Miao Z, Wolf FA *et al.* A test metric for assessing single-cell RNA-seq batch correction. *Nat Methods* 2019;16:43–9. https://doi.org/10.1038/s41592-018-0254-1
22. Korsunsky I, Millard N, Fan J *et al.* Fast, sensitive and accurate integration of single-cell data with Harmony. *Nat Methods* 2019;16:1289–96. https://doi.org/10.1038/s41592-019-0619-0
23. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;20:53–65. https://doi.org/10.1016/0377-0427(87)90125-7
24. Hubert L, Arabie P. Comparing partitions. *J Classif* 1985;2:193–218. https://doi.org/10.1007/BF01908075
25. Calle ML. Statistical analysis of metagenomics data. *Genom Inform* 2019;17:e6. https://doi.org/10.5808/GI.2019.17.1.e6
26. Zhang Y, Parmigiani G, Johnson WE. ComBat-seq: batch effect adjustment for RNA-seq count data. *NAR Genom Bioinform* 2020;2:lqaa078. https://doi.org/10.1093/nargab/lqaa078
27. Chapleur O, Madigou C, Civade R *et al.* Increasing concentrations of phenol progressively affect anaerobic digestion of cellulose and

associated microbial communities. *Biodegradation* 2016;27:15–27. https://doi.org/10.1007/s10532-015-9751-4
28. Schirmer M, Denson L, Vlamakis H *et al.* Compositional and temporal changes in the gut microbiome of pediatric ulcerative colitis patients are linked to disease course. *Cell Host Microbe* 2018;24:600–10. https://doi.org/10.1016/j.chom.2018.09.009
29. Zhang Y, Bhosle A, Bae S *et al.* Discovery of bioactive microbial gene products in inflammatory bowel disease. *Nature* 2022;606:754–60. https://doi.org/10.1038/s41586-022-04648-7
30. Hendriksen RS, Munk P, Njage P *et al.* Global monitoring of antimicrobial resistance based on metagenomics analyses of urban sewage. *Nat Commun* 2019;10:1124. https://doi.org/10.1038/s41467-019-08853-3
31. Hui HWH, Kong W, Goh WWB. Thinking points for effective batch correction on biomedical data. *Brief Bioinform* 2024;25. https://doi.org/10.1093/bib/bbae515
32. Faust K. Open challenges for microbial network construction and analysis. *ISME J* 2021;15:3111–8. https://doi.org/10.1038/s41396-021-01027-4
33. van de Velde C, Joseph C, Simoens K *et al.* Technical versus biological variability in a synthetic human gut community. *Gut Microbes* 2022;15:2155019. https://doi.org/10.1080/19490976.2022.2155019
34. Luecken MD, Gigante S, Burkhardt DB *et al.* Defining and benchmarking open problems in single-cell analysis. *Nat Biotechnol* 2025;43:1035–40. https://doi.org/10.1038/s41587-025-02694-w
35. Lloyd-Price J, Arze C, Ananthakrishnan AN *et al.* Multi-omics of the gut microbial ecosystem in inflammatory bowel diseases. *Nature* 2019;569:655–62. https://doi.org/10.1038/s41586-019-1237-9
36. Tang ZZ, Chen G. Zero-inflated generalized Dirichlet multinomial regression model for microbiome compositional data analysis. *Biostatistics* 2019;20:698–713. https://doi.org/10.1093/biostatistics/kxy025
37. Detlefsen NS, Hauberg S, Boomsma W. Learning meaningful representations of protein sequences. *Nat Commun* 2022;13:1914. https://doi.org/10.1038/s41467-022-29443-w
38. Yang L, Chen J. A comprehensive evaluation of microbial differential abundance analysis methods: current status and potential solutions. *Microbiome* 2022;10:130. https://doi.org/10.1186/s40168-022-01320-0

Received: December 1, 2025. **Revised:** February 13, 2026. **Accepted:** February 21, 2026