

Frontiers in Probability and the Statistical Sciences

Somnath Datta
Subharup Guha *Editors*

Statistical Analysis of Microbiome Data

Frontiers in Probability and the Statistical Sciences

Series Editors:

Frederi G. Viens

Department of Statistics and Probability

Michigan State University

East Lansing, MI, USA

Dimitris N. Politis

Dept Math, APM 5701

University of California, San Diego

La Jolla, CA, USA

Konstantinos Fokianos

Mathematics & Statistics

University of Cyprus Mathematics & Statistics

Nikosia, Cyprus

Michael Daniels

Department of Statistics

University of Florida

Gainesville, FL, USA

(Editor-in-Chief)

Somnath Datta

Department of Biostatistics

University of Florida

Gainesville, FL, USA

The “Frontiers” is a series of books (edited volumes and monographs) in probability and statistics designed to capture exciting trends in current research as they develop. Some emphasis will be given on novel methodologies that may have interdisciplinary applications in scientific fields such as biology, ecology, economics, environmental sciences, finance, genetics, material sciences, medicine, omics studies and public health.

More information about this series at <http://www.springer.com/series/11957>

Somnath Datta • Subharup Guha
Editors

Statistical Analysis of Microbiome Data



Springer

Editors

Somnath Datta
Department of Biostatistics
University of Florida
Gainesville, FL, USA

Subharup Guha
Department of Biostatistics
University of Florida
Gainesville, FL, USA

ISSN 2624-9987

ISSN 2624-9995 (electronic)

Frontiers in Probability and the Statistical Sciences

ISBN 978-3-030-73350-6

ISBN 978-3-030-73351-3 (eBook)

<https://doi.org/10.1007/978-3-030-73351-3>

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To our parents:
Sova and Sunil
Sunanda and Arup*

Preface

The role of microbiome in understanding human health is becoming increasingly evident. Besides medicine, microbiome-based studies revealing new insights are being conducted in various disciplines including dental research, agriculture, forestry, and so on. New sequencing technologies are producing large amounts of data leading to snapshots of microbial compositions in various scientific experiments. It is a non-trivial task to decipher the key microbiome signatures and reveal how they are influenced by various factors. Statisticians are playing an ever-increasing role in this important mission as evident from the numerous scientific publications and research presentations over the last decade. The current book provides a sample of the major topics in this important area of statistical research.

The idea of producing such a cohesive volume came to us about one and half years ago. After an initial discussion between the two of us, we approached Springer editor Laura Briskman, who showed her enthusiastic support toward this project. Subsequently, we approached a number of leading experts in this field and a great majority of them agreed to contribute a chapter in our book. We are grateful to all of them for providing us with high-quality material, and as a result, we are confident that this edited volume will serve as an important reference in this field. We ourselves, along with our doctoral students, have contributed a couple of chapters on topics not covered by others. All chapters have been refereed by a subject area expert and the two editors of this volume. We have attempted to make this potpourri of research topics as cohesive as possible and maintained a unified feel for the overall volume.

This volume should be useful for research statisticians looking to delve into a new area of methods, as well as applied research. In particular, doctoral students in statistics, biostatistics, and bioinformatics programs looking for dissertation research topics might also find this volume an important starting point. Upon our request, many of our contributors have provided scripts, R codes, and references to R packages for implementation of the analysis techniques and methods covered in the book. We hope this will make the book accessible and useful to many consulting biostatisticians working in academic and other settings dealing with microbiome data.

The book contains 13 chapters divided into five parts. Chapters in the first part discuss the sequencing technologies and bioinformatics pipelines leading to the microbiome abundance data which can be used for downstream statistical analysis. The opening chapter by Dorman et al. describes preprocessing techniques for the 16S rRNA amplicon sequencing technology. It describes the underlying statistical models behind each technique and concludes their chapter with a data visualization example. The next chapter by Li and Zheng considers the relatively newer shotgun sequencing and explains the underlying algorithms in detail for the popular bioinformatics tool Kraken. It also introduces the concepts of de Bruijn graph and the process of genome assembly. Overall, the chapter provides an excellent overview of dealing with shotgun metagenomic data. The following chapter by Anyaso-Samuel et al. takes a hands-on approach in illustrating various bioinformatics pipelines for shotgun sequencing by illustrating a metagenomics dataset using MetaPhlAn2, Kraken2, and Kaiju. In each case , it provides examples of the processing scripts which certain groups of readers will find very valuable. The latter part of that chapter illustrates the downstream analysis of building a statistical classifier with the resulting processed metagenomics data and makes a case for using an ensemble classifier rather than a single classifier.

The second part consists of two related chapters dealing with relative abundance data and how to conduct some basic exploratory analysis of the microbial communities. The chapter by Song and Sun introduces a number of distance/dissimilarity measures between a pair of microbiome samples. It then explains the technique for comparing two microbial communities using a phylogenetic tree and also introduces the concepts of the UniFrac distance and its variants. Overall, the chapter provides a comprehensive account of various methods for microbial community comparisons. Plantinga and Wu introduces the concepts of alpha and beta diversity measures and demonstrate how the later can be used to associate the microbiome signatures with a phenotype of interest. They describe visualization methods such as the principal coordinate analysis and a number of ordination plots. They conclude their chapter with a brief commentary to formal hypothesis testing in this context.

The four chapters of the third part present model-based techniques capable of handling the high-dimensionality, sparsity, and compositionality inherent in microbiome data. Martin, Uh, and Houwing-Duistermaat jointly model the relationships between repeated measurements on covariates and two sets of outcomes, namely, a continuous variable and the microbiome counts. The approaches rely on shared Gaussian random effects to model the correlation between the outcomes and account for overdispersion using a conjugate distribution, while offering insights into the complex longitudinal relationships of the data. Liu, Goren, Morris, Walker, and Wang carry out feature identification motivated by three biologically relevant questions: (1) which microbiome features are impacted by the treatments? (differential abundance analysis); (2) which features modify or influence the treatment effect on the outcome of interest? (mediation analysis); and (3) after adjusting for confounders, which features are potentially causally associated with outcome? The

next chapter by Wang and Zhao reviews cutting-edge methodologies that tackle the challenges of widely varying library sizes across microbiome samples drawn from a fraction of the original ecosystem and sparse abundance counts for a large number of taxa. The tree-structured phylogeny of the taxa is incorporated into empirical Bayes estimation of relative abundances, regularization-based subcomposition selection, and variable fusion in regression models with compositional predictors. The chapter by Zhao and Satten covers strategies for accounting for the ubiquitous biases in the relative abundances that are contributed by the various steps of the experimental and analysis pipeline. The chapter presents a log-linear model for quantifying bias in model community data and beyond. The model facilitates testing complex hypotheses through permutation-based F-tests and accommodates designs where the samples differ in the number of bacteria.

The fourth part has two chapters that focus on Bayesian approaches for microbiome data analysis. Koslovsky and Vannucci cover Bayesian models for integrative analyses combining microbiome data with other available information to identify significant associations between taxa and a set of predictors. They describe hierarchical Dirichlet-multinomial (DM) and Dirichlet-tree multinomial (DTM) regression models with spike-and-slab priors for detecting significant associations. Strategies for inclusion indicators using DM and incorporating the phylogenetic structure using DTM models are discussed. The next chapter, by Guha and Datta, proposes an approximate singular value decomposition of the abundance matrix to restore, via the Bayesian paradigm, the duality between orthonormal vectors associated with pairwise distances between the sample units (such as UniFrac) and orthonormal vectors of the operational taxonomic units (OTUs). The approach provides inferences beyond point estimates, such as standard errors and credible intervals, and for arbitrary functionals of interest, such as the contributions of individual OTUs.

The fifth part consists of special topic chapters. The chapter by Lu and Ishwaran discusses methods for paired microbiome samples collected from two locations of the same individual or from two individuals with family ties. Applying ideas from classification tree splitting, it proposes a novel approach based on the Gini split-statistic that disentangles different types of associations, such as host genotype and environmental exposure effects. Following this, Ma, Yue, and Shojaie review established techniques for inferring microbial interaction networks from microbial abundance data. Based on both marginal and conditional associations, the methods are robust to the spurious correlations resulting from compositionality and seek to discover the true underlying network structure. The chapter presents a comprehensive empirical evaluation using simulated data sets.

Once again, we are sincerely grateful to the exceptional researchers for their invaluable contributions. We appreciate their inventiveness, enthusiasm, and hard work, and their willingness to make the revisions that we suggested. Reading the authors' outstanding contributions has greatly enhanced our own understanding of

microbiome data analysis, and we hope most, if not all, readers will similarly profit from this book.

Gainesville, FL, USA

Somnath Datta

Gainesville, FL, USA

Subharup Guha

January 2021

Acknowledgments

We thank Springer and the series editors for accepting our book proposal and all the contributors for their hard work for timely delivery of their high-quality chapters and subsequent revisions.

We also thank the entire Springer team for their part, especially Laura, Kirthika, and Gomathi. We thank our doctoral students, Archie Sachdeva and Samuel Anyaso-Samuel, for compiling the whole volume.

Contents

Part I Preprocessing and Bioinformatics Pipelines

Denoising Methods for Inferring Microbiome Community Content and Abundance	3
Karin S. Dorman, Xiyu Peng, and Yudi Zhang	
Statistical and Computational Methods for Analysis of Shotgun Metagenomics Sequencing Data	27
Hongzhe Li and Haotian Zheng	
Bioinformatics Pre-Processing of Microbiome Data with An Application to Metagenomic Forensics	45
Samuel Anyaso-Samuel, Archie Sachdeva, Subharup Guha, and Somnath Datta	

Part II Exploratory Analyses of Microbial Communities

Statistical Methods for Pairwise Comparison of Metagenomic Samples ..	81
Kai Song and Fengzhu Sun	
Beta Diversity and Distance-Based Analysis of Microbiome Data	101
Anna M. Plantinga and Michael C. Wu	

Part III Statistical Models and Inference

Joint Models for Repeatedly Measured Compositional and Normally Distributed Outcomes	131
Ivonne Martin, Hae-Won Uh, and Jeanine Houwing-Duistermaat	
Statistical Methods for Feature Identification in Microbiome Studies	175
Peng Liu, Emily Goren, Paul Morris, David Walker, and Chong Wang	
Statistical Methods for Analyzing Tree-Structured Microbiome Data	193
Tao Wang and Hongyu Zhao	

A Log-Linear Model for Inference on Bias in Microbiome Studies	221
Ni Zhao and Glen A. Satten	

Part IV Bayesian Methods

Dirichlet-Multinomial Regression Models with Bayesian Variable Selection for Microbiome Data	249
Matthew D. Koslovsky and Marina Vannucci	

A Bayesian Approach to Restoring the Duality Between Principal Components of a Distance Matrix and Operational Taxonomic Units in Microbiome Analyses	271
Subharup Guha and Somnath Datta	

Part V Special Topics

Tree Variable Selection for Paired Case–Control Studies with Application to Microbiome Data	295
Min Lu and Hemant Ishwaran	

Networks for Compositional Data	311
Jing Ma, Kun Yue, and Ali Shojaie	

Index	337
--------------------	------------

Part I

Preprocessing and Bioinformatics Pipelines

Denoising Methods for Inferring Microbiome Community Content and Abundance



Karin S. Dorman, Xiyu Peng, and Yudi Zhang

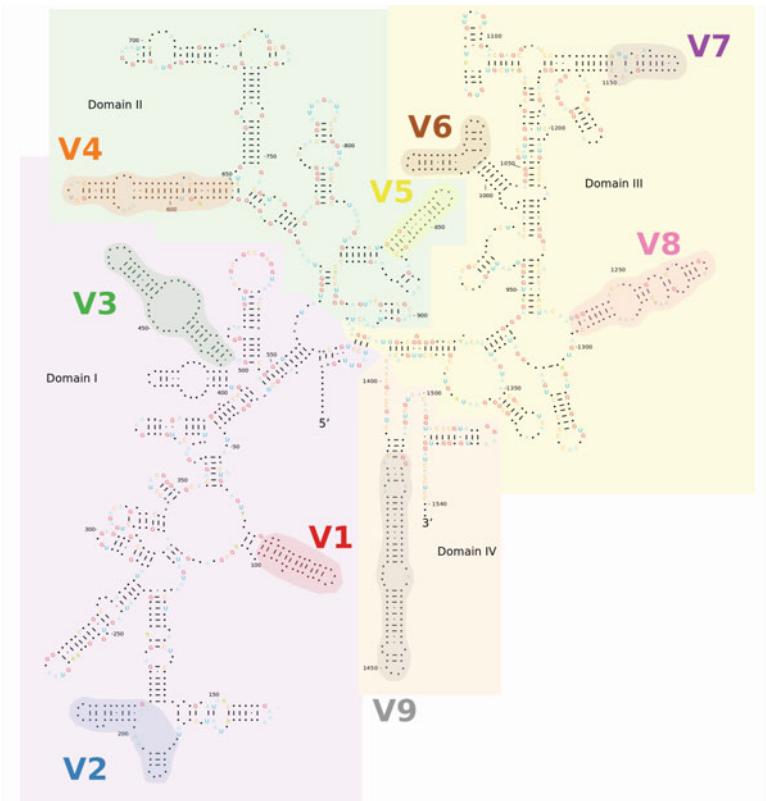
1 Introduction

High-throughput sequencing has revolutionized the study of microbial communities. With up to millions of reads of short DNA sequence fragments per sample, microbiologists are now able to investigate the composition and dynamics of complex natural (uncultured) microbial communities to answer questions related to human health [29, 51, 69, 80] and ecology [5, 13, 54, 82]. The analysis of microbial communities usually begins with an identification of community members and their abundance, but the task is challenging because natural community diversity is obscured by the biases and errors of library preparation and subsequent sequencing [25, 32, 40]. Chapter 2 will discuss methods related to shotgun metagenomics. This chapter focuses on methods for amplicon sequencing.

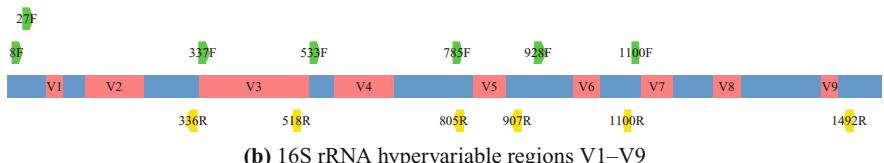
Amplicon, or biomarker, sequencing amplifies and sequences biomarker genes, like the 16S rRNA gene or the fungal internal transcribed spacer (ITS), to identify and quantify the organisms in a community. These biomarkers are highly conserved, slowly evolving genes that persist widely across the tree of life, yet they also contain *hypervariable* regions with nucleotide differences that can be used to identify most genera, many species, and some strains [37, 53] (Fig. 1).

While seemingly an ideal fingerprinting device, the amplicon approach suffers from several technical difficulties [14, 75]. Some artifacts affect all sequencing-based metagenomics approaches. For example, there is demonstrated sensitivity to sample storage or DNA extraction method [18, 50, 71] and a smaller effect of DNA sequencing platform [59, 79]. However, amplification by polymerase chain

K. S. Dorman (✉) · X. Peng · Y. Zhang
Iowa State University, Ames, IA, USA
e-mail: kdorman@iastate.edu; xiyupeng@iastate.edu; yudiz@iastate.edu



(a) 16S rRNA secondary structure



(b) 16S rRNA hypervariable regions V1–V9

Fig. 1 *Escherichia coli* 16S rRNA (a) secondary structure and (b) primary structure showing variable regions V1–V9 and the location of some commonly used universal primers

reaction (PCR) is a major distorting force specific to biomarker sequencing. The choice of primers strongly affects sample composition [30], missing some species entirely [26] and distorting abundance estimates of the detectable species [79]. Even a single nucleotide mismatch to the primers can affect abundance estimates [2], but there are a myriad of other amplification biases, from polymerase choice, number of PCR cycles, GC content to secondary structure [31, 77]. Years of progress have lead to a growing consensus on best amplification practices that can reduce [30, 52, 86] but not eliminate biases (see Chap. 9). Fortunately, despite

the biases, biomarker sequencing can provide quantitative information useful for assessing how communities change in response to experimental and environmental perturbations [15, 83]. Furthermore, it is hard to beat the price of amplicon sequencing, especially for complex communities [13].

To extract the necessary quantitative information from an amplicon sequencing dataset, it is important to identify the true sequences among the errored sequences produced by amplified PCR errors and the elevated error rates of high-throughput sequencing. It is also important to reproducibly quantify the abundance of each true sequence, albeit with biases introduced during library preparation prior to sequencing. When modern amplicon sequencing first emerged, errors generated during PCR and sequencing were removed by grouping reads into operational taxonomic units (OTUs) [12, 74]. OTUs were constructed based on an empirical sequence similarity threshold, usually 97% [44, 76]. Taxonomic labels could then be assigned to OTUs and their consensus sequences and read counts passed to downstream statistical and phylogenetic analyses.

OTU-based methods have fallen out of vogue since it has been recognized that amplicon sequencing data from current Illumina platforms contain enough information to support *de novo* single-nucleotide sequence resolution [9]. Instead of setting an arbitrary similarity threshold for grouping reads, new methods consider both sequence similarity and abundance to resolve reads into clusters representing amplicon sequence variants (ASVs) [1, 9, 23, 27, 33, 61, 78]. These methods have been called *denoisers* because they “remove” technical errors in the observed reads. Their aim is to identify true biological sequences and their surrounding clusters of errored reads, not the consensus or centroid sequences of arbitrary sequence groupings. Denoising methods are now the recommended first step in biomarker gene analysis because of their high resolution, low false positive rate, and cross-sample consistency [8, 43].

In this chapter, we discuss methods for denoising microbiome amplicon data. We briefly summarize algorithmic methods that do not formulate explicit probability models in Sect. 2, but this chapter is primarily focused on probabilistic denoising approaches for ASV discovery covered in Sect. 3. The algorithmic methods include OTU-based methods that group reads by pairwise distances and the algorithmic denoisers that also consider relative abundances. The algorithmic denoisers can be thought of as approximations to probabilistic denoisers, with error parameters hard-coded or chosen by the user. In Sect. 4, we address the problem of assessing method performance, before ending with some brief conclusions in Sect. 5.

2 Common Algorithmic Denoising Strategies

We will start with OTU-based clustering methods, before examining the newer denoising (ASV-based) methods for identifying real biological sequences. OTU-based clustering methods can be roughly divided into two categories: reference-based methods (including both closed and open references) and reference-free (*de*

novo) methods. The most well-known OTU-based clustering methods have been integrated into QIIME/QIIME2 [7, 11] and Mothur [74], two commonly used software packages for microbiome data analysis.

Reference-based clustering methods cluster reads against a reference database [57]. Reads sufficiently similar to a reference sequence in the database will be clustered into the same OTU. If the same database is used, OTU assignments from closed-reference clustering methods are consistent, thus comparable across studies. However, they fail to correctly cluster reads if any of the biological source sequences, or another very similar sequence, is not included in the reference database, since reads that fail to match any reference sequence are discarded. Unfortunately, reference databases are far from perfect. They are incomplete [68] and contain many mislabeled sequences [24, 45]. In order to overcome the incompleteness of databases, open-reference clustering methods perform *de novo* clustering (see below) on sequences that do not map to the reference database [57]. Not surprisingly, all methods that rely on a reference database are sensitive to the database they use and tend to generate more false positives than reference-free methods [84].

Most *de novo* algorithms cluster reads into OTUs based on their pairwise sequence similarities [57]. Hierarchical, more specifically agglomerative, clustering is commonly used in *de novo* algorithms [74]. All hierarchical methods require pairwise distances between the reads and a definition of inter-cluster distance, either single-linkage, complete-linkage, or average-linkage clustering. Each linkage type compares pairs of sequences, one from each cluster, to a user-specified threshold. Complete-linkage clustering merges two clusters when *all* pairs of sequences are closer, single-linkage clustering when there exists *a* pair of sequences closer, and average-linkage clustering when *average* pairwise distances are closer than the threshold [55]. All methods are greedy, merging the most similar reads or clusters first.

The disadvantage of agglomerative clustering methods is the need to compute all pairwise distances between reads. Thus, the computational cost increases quadratically with the number of unique sequences. To reduce the computational complexity for large-scale sequencing data, greedy heuristic algorithms are proposed to approximate hierarchical clustering [21]. One popular greedy *de novo* clustering strategy is UPARSE [22]. It considers, in order of decreasing abundance, unique sequences observed among the reads as candidate cluster centroids. Either the candidate is merged with an existing OTU if the sequence similarity is above the threshold, or it is designated the centroid of a new OTU. The process continues until all unique sequences above a minimal abundance threshold have been processed.

The two most popular algorithmic, non-probabilistic denoising methods are UNOISE2 [23] and Deblur [1]. Normally, Deblur uses a reference database to both pre-filter reads and post-filter discovered ASVs, but it can also run in reference-free *de novo* mode. Both methods consider the relative abundance of sequences, in addition to their similarity, when deciding to merge clusters. UNOISE2 is very similar to UPARSE. After sorting all unique sequences, s_1, s_2, \dots , in order of decreasing abundance, a_1, a_2, \dots , UNOISE2 considers what to do with the next

most abundant sequence s_j with abundance a_j . Either s_j will get assigned to the closest cluster i with

$$\frac{a_j}{a_i} \leq \frac{1}{2^{\alpha_u d_L(s_i, s_j) + 1}}, \quad (1)$$

where $d_L(s_i, s_j)$ is the Levenshtein distance [56] between unique sequences s_i and s_j , and a_i will be incremented by the count a_j , or if there is no such cluster i , then sequence s_j will become the centroid of a new cluster. Formula (1), including parameter α_u , was learned from several mock and real Illumina datasets. Deblur also considers unique sequences in abundance order but makes decisions based on estimates of *true* abundance a_{it} . After aligning all unique sequences s_i with abundance $a_i > 1$ using MAFFT [39] and initializing $a_{it} = a_i$, it then processes all reads in abundance order. For the i th unique sequence, the true abundance $a_{it} \approx \frac{a_{it}}{1-\alpha_d}$, increased by the fraction α_d of misreads that contain at least one error. Then, the abundance of all less abundant unique sequences with $j > i$ is reduced $a_{jt} = a_{jt} - \beta [d_H(s_i, s_j)] a_{it}$ by the expected number of misreads from true sequence s_i . In these equations, $d_H(s_i, s_j)$ is the Hamming distance [56] between sequences s_i and s_j and $\beta(d)$ is an empirically estimated probability that s_i is misread as s_j , when they have d differences. If $a_{it} < 0$, s_i is presumed to be an error sequence.

The key assumptions underlying the algorithmic denoising methods are that all true sequences are multiply observed among the reads without errors and most misreads are sourced from similar, but more abundant sequences in the dataset. Thus, a unique sequence is more likely to be a true sequence if it is abundant and distant from other abundant sequences. The two most popular methods also learn several algorithmic run parameters from real Illumina datasets, so they assume that these parameters are shared across platforms, labs, and samples. The statistical methods we consider next build on the logic of the algorithmic denoisers but incorporate more flexibility in their error models.

3 Model-Based Denoising

There is an error model underlying both popular algorithmic denoisers, UNOISE2 and Deblur, but neither fully formulates it before converting it into a well-tuned algorithm. We now discuss methods that fully formulate a probabilistic error model. The methods designed for the Illumina platform (DADA2 and AmpliCI) make use of *quality scores* [28], which are discretized probabilities accompanying each read nucleotide, roughly communicating the probability of a sequencing error at that position. Because of the massive data and concomitant computational challenges, each of these approaches ultimately utilizes approximations to estimate parameters and conduct model selection for the number of true sequences. Even so, these methods tend to be slower than the algorithmic approaches, especially UNOISE2,

which is resoundingly memory and time efficient [61]. However, because the model-based approaches explicitly reveal their assumptions, it is often easier to identify the weaknesses and speculate on possible future improvements.

3.1 Hierarchical Divisive Clustering

DADA2 [9] proposes a divisive clustering algorithm with a Poisson model to partition clusters. It first partitions the read data into unique sequences and then assumes reads matching unique sequence s_i either are all error-free or all share the same error(s). Like the algorithmic denoisers, DADA2 sorts all unique sequences, s_1, s_2, \dots , by decreasing abundances a_1, a_2, \dots . Initially, DADA2 assumes a single cluster with true ASV $\mathbf{h}_1 = s_1$, the most abundant unique sequence. It then iteratively splits off a new cluster around unique sequence s_i if its abundance a_i is unusually high given its current cluster. Specifically, if sequence $s_i \neq \mathbf{h}_k$ is in cluster k , DADA2 assumes that the number of misreads of \mathbf{h}_k with sequence s_i follows a Poisson distribution,

$$p_{\text{Pois}}(a; n_k \lambda_{ki}) = \frac{e^{-n_k \lambda_{ki}} (n_k \lambda_{ki})^a}{a!},$$

where n_k is the number of reads in cluster k and λ_{ki} is the per-read rate at which true sequence \mathbf{h}_k produces sequence s_i by misread. If errors are independent at sites in the read, then λ_{ki} is the product over the l aligned nucleotides,

$$\lambda_{ki} = \prod_{j=1}^l \Pr(s_{ij}; h_{kj}, q_{ij}), \quad (2)$$

where $\Pr(s_{ij}; h_{kj}, q_{ij})$ is the probability that original nucleotide h_{kj} is read as nucleotide s_{ij} with quality score q_{ij} at the aligned position j . The quality score q_{ij} associated with position j of sequence s_i is an average of the observed quality scores at position j of all reads matching sequence s_i . The validity of the null hypothesis that all reads of sequence s_i are misreads of true sequence \mathbf{h}_k is evaluated with the probability of observing a_i or more reads of sequence s_i given that there was at least one observation of s_i , i.e., the p -value,

$$\Pr(a \geq a_i \mid a_i > 0; \lambda_{ki}, n_k) = \frac{1}{1 - p_{\text{Pois}}(0; n_k \lambda_{ki})} \sum_{a=a_i}^{\infty} p_{\text{Pois}}(a; n_k \lambda_{ki}). \quad (3)$$

If the p -value falls below a user-settable threshold, a new partition i is formed with s_i as its center. DADA2 continues partitioning clusters until there are no more unusually abundant unique sequences.

In recent comparisons of the 16S rRNA amplicon denoising pipelines, DADA2 proved to have greater sensitivity, with slightly reduced precision, compared to Deblur and UNOISE2 [58, 63]. A problem with the DADA2 approach is the compression of reads into unique sequences, the failure to use original quality score information, and the treatment of all sequence-matched reads as cohesive groups. If multiple true sequences produce the same misread sequence, DADA2 may incorrectly detect the misread as unusually abundant since it assumes that all these misreads arose from the same source sequence. Another problem is that DADA2 fails to account for biological variants with true insertion or deletion (indel) differences, so some true indel variants are likely to be missed [33]. Finally, the Poisson model does not account for the overdispersion of read data, for example, caused by PCR amplification [6], which results in a higher-than-expected variance when generating errors [62]. Like UNOISE2 and Deblur, which openly discuss the issue [1] (and see the Discussion section in [61]), DADA2 compensates for unmodeled errors (PCR and contamination) by using a very conservative criterion (10^{-40}) on the p -values (3) when partitioning clusters. Of course, some true sequences near other abundant true sequences are likely to remain undetected with such conservative decisions.

3.2 Finite Mixture Model

AmpliconNoise [65] (or PyroNoise [64]) was the first finite mixture model proposed for correcting sequencing errors. They modeled 454 pyrosequencing, a high-throughput technology that detects nucleotides as they are incorporated during complementary DNA strand synthesis from a sampled template strand. The 454 technology has since been discontinued, but we briefly discuss the statistical method for its historical significance.

Pyrosequencing raw data are flowgrams, which for each cycle of T, A, C, and G through the instrument, record the fluorescent intensity when complementary nucleotides are incorporated in the DNA synthesis reaction (Fig. 2). The intensity fluctuates around discrete intensity values, increasing from lower levels when no nucleotides are incorporated, i.e., when the cycle nucleotide is not complementary to the template nucleotide, to higher levels when one nucleotide is incorporated, to even higher levels when the cycle nucleotide is complementary to a homopolymer run. The model assumes that flowgrams are independently generated from a mixture model with K components, and the likelihood of dataset \mathcal{F} of n flowgrams is

$$L(\theta | \mathcal{F}) = \prod_{i=1}^n \sum_{k=1}^K \pi_k \mathcal{J}(f_i; \mathbf{h}_k), \quad (4)$$

where π_k is the relative abundance of the k th component and $\mathcal{J}(f_i; \mathbf{h}_k)$ is the joint density of observed flowgram f_i given true sequence \mathbf{h}_k . The flowgrams are

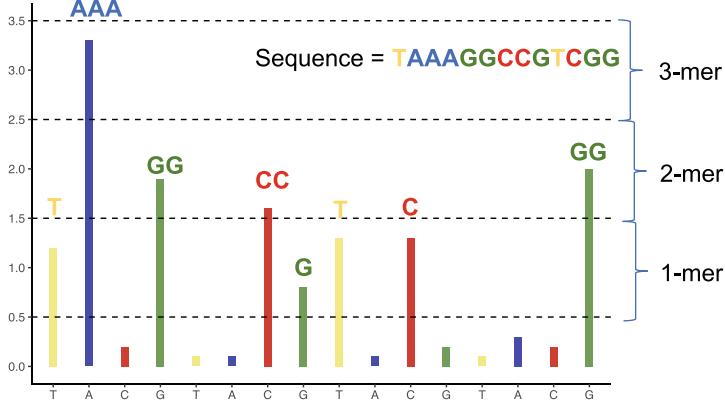


Fig. 2 454 pyrosequencing flowgram

assumed to be generated as exponentially decaying functions of their distance to the perfect noise-free flowgram \mathbf{u}_k of sequence \mathbf{h}_k ,

$$\mathcal{J}(f_i; \mathbf{h}_k) = \frac{\exp(-d(f_i, \mathbf{u}_k)/\sigma)}{\sigma}, \quad (5)$$

where the distance

$$d(f_i, \mathbf{u}_k) = \frac{1}{M} \sum_{j=1}^M \{-\log f(f_{ij}; u_{kj})\}, \quad (6)$$

for $f(f_{ij}; u_{kj})$, the density of signal f_{ij} from homopolymer u_{kj} at cycle j of M cycles (the number of homopolymers in sequence \mathbf{u}_k).

An expectation–maximization (EM) algorithm is developed to maximize (4) and infer the true sequences, $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_K$, and their relative frequencies, $\pi_1, \pi_2, \dots, \pi_K$. To initialize the EM algorithm, a complete-linkage hierarchical clustering method with a given distance cutoff is performed to form an initial partition. The results of AmpliconNoise may depend on the quality of its initialization [65], since the EM algorithm finds a local optimum.

AmpliCI [61] formulates a finite mixture model for denoising Illumina amplicon sequencing data. Instead of flowgrams, AmpliCI clusters reads with observed quality scores. It assumes that reads are independently generated from a K -component mixture distribution, where the k th component generates the reads and misreads of true ASV sequence \mathbf{h}_k . Like DADA2, AmpliCI takes into account quality scores, but unlike DADA2, it does not average them across reads with the same sequence. The likelihood function of the read set $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n\}$ is

$$L(\theta | \mathcal{R}) = \prod_{i=1}^n \sum_{k=1}^K \pi_k \Pr(\mathbf{r}_i; \mathbf{h}_k, \mathbf{q}_i), \quad (7)$$

where π_k is the relative abundance of the k th true sequence and \mathbf{q}_i are the quality scores for the i th read. Assuming errors are independent across sites, the conditional probability of read \mathbf{r}_i given the true ASV \mathbf{h}_k is

$$\Pr(\mathbf{r}_i; \mathbf{h}_k, \mathbf{q}_i) = \Pr(d_i; \mathbf{h}_k) \prod_{j=1}^{l_k} \Pr(r_{ij}; h_{kj}, q_{ij}),$$

where $\Pr(d_i; \mathbf{h}_k)$ is the probability of d_i observed indel events in the i th read and $\Pr(r_{ij}; h_{kj}, q_{ij})$ is the substitution probability for generating read nucleotide r_{ij} from true nucleotide h_{kj} at the j th aligned read nucleotide (1 if there is a deletion) to the l_k positions in the k th true ASV \mathbf{h}_k . Given indel errors are rare in Illumina data, the model assumes that d_i is approximately modeled as a truncated Poisson with mean $l_k \delta$, where δ is the (known) indel error rate. The substitution probabilities are parameterized using LOESS regression, a model borrowed from DADA2, and they are fit alternating with ASV selection until the estimates stabilize. Then, the fitted error profile is used to restart the algorithm and select ASVs until no more qualified candidates remain.

AmpliCI formalizes the approximations of Deblur to estimate the true abundances into a novel greedy algorithm to rapidly select sequences and approximately maximize the mixture model (7). To avoid false positives, the method insures that the model fit has improved with every added sequence by computing an approximate Bayesian information criterion (BIC). It screens for possible contaminants by computing a diagnostic probability, similar to DADA2's p -value. It overcomes DADA2's loss of quality score information, but like DADA2, it does not consider the overdispersion of count data, makes conservative decisions to overcome error model misspecification, and overestimates the error rates.

3.3 Denoising Long-Read Technology

The denoising methods we have described are designed for Illumina amplicon data and will not work for long-read technologies [67]. The existing methods are not applicable because indels are common and read lengths are highly variable in the newer technology, both issues assumed to be negligible for Illumina read data. In fact, sequencing error rates are very high in long-read technologies. Oxford Nanopore Technology has a 5–25% error rate [85], and Pacific Bioscience has a 13% error rate [3]. Current denoising methods for long reads [10, 46] are mainly designed for circular consensus sequences (CCS), where a DNA molecule is circularized and read multiple times before reporting a consensus sequence [34]. The CCS approach can dramatically reduce error rates, making them comparable to those of short-read technology [34] and producing ~50% error-free full-length 16S reads in Pacbio CCS data [10]. However, for longer targeted gene sequences, mean error rates exceed 2% due to insufficient coverage, producing far less error-free

reads [38]. With higher error rates, abundance-based denoising methods lose power to detect low abundance variants. Recently, unique molecular identifiers [42] have been applied to long reads [38], providing new opportunities to correct errors in noncircular long reads. Since long reads increase the resolution of biomarker studies and bypass the need for, and consequent bias of, PCR amplification, it is likely that there will be continued development in denoising methods for long reads.

4 Model Assessment

When there are competing methods and models, it is important to be able to make fair, accurate, and extensive comparison of the approaches. Comparing denoisers has proven to be particularly difficult. Most denoisers are an integral part of a complete amplicon processing pipeline that includes read filtering and trimming, denoising, chimera detection, and other post-processing [63], so it is difficult to isolate the effect of the denoiser [61]. Worse yet, simulation is not yet capable of replicating the vagaries of real data [61], and though there has been heavy use of mock datasets [9, 19, 30, 48, 58, 60, 77], the truth is not always clear for mock communities, as we shall see below.

4.1 With Known Truth

Both simulated and mock datasets are used for benchmarking with a known truth. Mock datasets are datasets generated from real samples of known microbial communities, for which reference sequences are provided. Mock data reflect true error properties of PCR and sequencing much better than simulated data. Since the true classification of reads is not available for mock datasets, algorithms are often only evaluated for their ability to recover true sequences in the reference set, but the reference set may not be correct. Contamination is common in amplicon sequencing, especially when amplifying from low template concentrations [41, 72, 87]. Furthermore, it is entirely plausible, especially at greater sequencing depths, that additional sequence variants will be discovered that are missing from the reference list.

4.1.1 Accuracy in ASV Identification

The ability to accurately identify the number and identity of ASVs is key to the taxonomic profiling of microbial communities. To assess the ability of methods to recover ASVs, *recall* (proportion of recovered ASVs among true ASVs) and *precision* (proportion of true ASVs among predicted ASVs) are commonly used for evaluation. All denoising methods have user-controlled run parameters that affect the precision–recall trade-off.

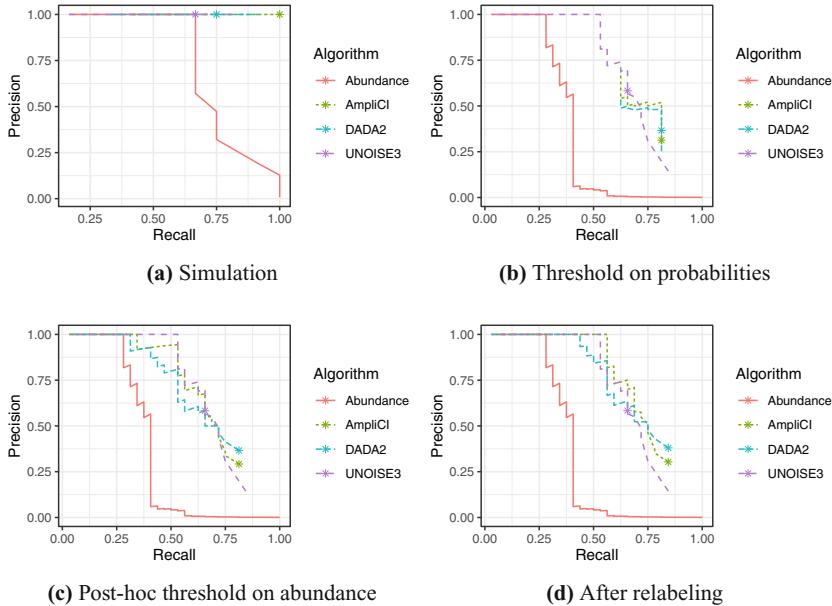


Fig. 3 Precision–recall (PR) curves for (a) a simulated dataset and (b)–(d) a mock community. In (b), the p -value for DADA2 and the diagnostic probability for AmpliCI are varied. In (c) and (d), the default output of DADA2 and AmpliCI is post-processed and thresholded on abundance. In (d), one likely false positive observed more than 2500 times is relabeled as a true positive. “Abundance” is the baseline method, where sequences are selected as ASVs in order of decreasing abundance. Results under the default parameter settings for each method are shown in (a)–(d) as “**”; The abundance method has no “**” since there is no universal recommendation for setting the abundance threshold

Precision–recall (PR) curves built for simulated or mock communities can help illustrate the effect of such run parameters. One naive and commonly used run parameter is a threshold on ASV (cluster) abundance: the more times a sequence is observed among the reads, the more likely it is a true ASV. Most algorithms do not consider singletons as valid clusters, but many recommend much higher thresholds. For example, UNOISE3 currently recommends a minimum abundance of eight [20]. DADA2 and AmpliCI recommend setting a low abundance threshold and instead imposing a threshold on the p -value (DADA2) and the diagnostic probability (AmpliCI), generically probability, for screening new ASVs. We run DADA2 v1.16.0 with different p -value thresholds, AmpliCI v1.0 with different diagnostic probability thresholds, and UNOISE3 v11.0 with different abundance thresholds on two datasets. We first examine an unbalanced simulated dataset of 3000 reads from 12 ASVs with two to five nucleotide differences and one 4-nucleotide (nt) deletion. Regardless of threshold, DADA2 always misses the indel ASV and UNOISE3 always misses two additional ASVs in the simulated dataset, and both DADA2 and UNOISE3 miss additional ASVs at their default settings (Fig. 3a). We also examine the PR curve for the Extreme mock dataset [9], which

is extremely unbalanced, with observed abundance (the number of error-free reads) ranging from just 2 to 275,000. The PR curves show that the probabilistic models can eke out higher recall with a cost in precision and that both DADA2 and AmpliCI default parameter settings are not optimally tuned for these examples (Fig. 3b).

For these examples, thresholding on probability fails to resolve the upper left portion of the PR curve for DADA2 and AmpliCI because these probabilities round to 0 for the high abundance true positives revealed by this portion of the curve. Figure 3c and d examines the PR curves when DADA2 and AmpliCI ASVs are *post hoc* thresholded on observed sequence abundance. The curves are inferior in the lower right, compared to thresholding on probabilities, but they now extend all the way to perfect precision, where UNOISE3 continues to dominate the probabilistic methods. One of the false positives is a high abundance sequence that is observed 2500 times among the reads. It involves a single A to G transition relative to one of the reference sequences published with the Extreme mock dataset, and it is identical to a sequence in the NCBI nucleotide database. If we relabel this ASV as a true positive, the curves in Fig. 3d result, where AmpliCI dominates the other methods throughout the upper left portion of the PR curve.

In summary, the probabilistic methods DADA2 and especially AmpliCI, tend to do well in simulation [61], but the picture is far less clear when analyzing mock datasets, where there remains uncertainty about which discovered sequences are actual true positives. Confirmed by others for DADA2 [58, 63], the probabilistic methods are able to eke out a bit more sensitivity in some areas of the PR curve, which is exactly what they are designed to do. However, their use of misspecified and poorly estimated error models [61] forces them to use overestimated error rates, which while helping them avoid excess false positives severely limits their ability to further improve the precision–recall trade-off. They are also limited by their failure to acknowledge stochasticity or bias in PCR [6]. Error sequences that are randomly or deterministically, by PCR bias, over-amplified will exceed a threshold placed on the probability, while true sequences that are under-amplified will be eliminated by the threshold. It is clear that further improvement will only come when the problem of errors that leave no trace in the quality scores, i.e., PCR errors, PCR bias, and contamination, is tackled.

4.1.2 Accuracy in Read Assignments

If true clustering labels are provided, for example, in simulated datasets, several metrics for assessing clustering methods can be applied to assess the performance of denoising algorithms on read assignments.

The adjusted Rand index (ARI) [35] is the adjusted-for-chance version of the Rand index (RI) [66] for comparing the predicted clustering with the true clustering. Let $X = \{X_i\}$ be a partition (an exhaustive collection of nonoverlapping subsets) induced by the solution of a clustering algorithm and $Y = \{Y_j\}$ the partition induced by the true clustering labels of the same set of n objects. The size of the overlap between set X_i and set Y_j is $n_{ij} = |X_i \cap Y_j|$, and the ARI is defined as

$$\text{ARI} = \frac{\sum_{i,j} \binom{n_{ij}}{2} - [\sum_i \binom{|X_i|}{2} \sum_j \binom{|Y_j|}{2}] / \binom{n}{2}}{\frac{1}{2} [\sum_i \binom{|X_i|}{2} + \sum_j \binom{|Y_j|}{2}] - [\sum_i \binom{|X_i|}{2} \sum_j \binom{|Y_j|}{2}] / \binom{n}{2}}. \quad (8)$$

The ARI achieves a maximum score of 1 when the predicted clustering is perfectly aligned with the true clustering, while a random clustering is expected to have an ARI equal to 0.

The V-measure [70] is an entropy-based method for evaluating clustering performance. It has two components, completeness c and homogeneity h , defined as

$$c = 1 - H(X | Y) / H(X), \quad (9)$$

$$h = 1 - H(Y | X) / H(Y), \quad (10)$$

where $H(\cdot)$ and $H(\cdot | \cdot)$ denote the entropy and conditional entropy functions. Specifically, $H(Y | X)$ and $H(Y)$ are defined as

$$H(Y | X) = - \sum_{i=1}^{|X|} \sum_{j=1}^{|Y|} \frac{n_{ij}}{N} \log \frac{n_{ij}}{\sum_{i=1}^{|Y|} n_i},$$

$$H(Y) = - \sum_{j=1}^{|Y|} \frac{\sum_{i=1}^{|X|} n_{ij}}{N} \log \frac{\sum_{i=1}^{|X|} n_{ij}}{N}.$$

Homogeneity achieves its maximum value if all clusters contain only observations belonging to a single true class. Completeness achieves its maximum value if all the observations of a given true class are assigned to the same cluster; assigning all data points to one cluster is one way to achieve perfect completeness. The V-measure is defined as the harmonic mean of homogeneity and completeness,

$$V = \frac{2hc}{h+c}, \quad (11)$$

similar to the way precision and recall are combined into the F -score for the binary classification problem. We illustrate the usage of the ARI and V-measure for method comparison in Sect. 4.2.1.

4.2 With Unknown Truth

Assessing performance on real datasets is always challenging, since there are no labels, but plenty of noise (chimeras and contaminants). In order to assess clustering performance, some methods may create a “true” clustering solution based on super-

vised classification, aligning reads to known reference databases [9, 23, 58, 61]. However, as discussed previously, the current 16S rRNA gene databases are far from perfect, having been built from previous OTU-based analyses [24, 84]. The provided “truth” will ignore any biological variants that are not already in the reference databases, and previously deposited error sequences will validate the same errors in the real dataset. Since Illumina sequencing errors do display patterns [73], it is likely that reference-based truths will propagate existing errors without providing good metrics for performance evaluations. Below we propose several strategies that could be used for evaluation of denoisers when the truth is not known.

4.2.1 Assessment with UMIs

Some amplicon datasets provide technical “sequence” labels, which enable higher resolution to detect rare variants. These labels are short random sequences, called unique molecular identifiers (UMIs) [42] or primer IDs [36], attached to sample sequences before PCR amplification. UMIs offer an opportunity to evaluate a proposed clustering since reads with the same UMI should be in the same cluster except when there is UMI reuse. Treating UMIs as true class labels, the completeness of the V-measure is a useful criterion for evaluation, since it only penalizes splitting of UMIs across clusters but does not penalize multiple UMIs in the same cluster, which are expected for highly abundant ASVs. Ideally, if there are no errors in the UMIs and no chimeras generated during PCR, the completeness should achieve the value one.

We compare DADA2 and AmpliCI denoising methods on a dataset (SRR2241783) of HIV *env* amplicon sequences with UMIs. UNOISE3 is not compared, since final reads are not assigned under its denoising model. DADA2 identifies 88 clusters and AmpliCI identifies 45 clusters. The completeness of DADA2 is 0.75, lower than 0.80, the completeness of AmpliCI. The overall V-measure for DADA2 (0.49) is slightly higher than AmpliCI (0.48), since the homogeneity for the DADA2 solution is higher (0.36 vs 0.34). DADA2 also has slightly higher ARI (0.0334 vs 0.0326). AmpliCI may be underestimating the number of clusters in this dataset, but DADA2 is more likely to split reads with the same UMI into different clusters.

4.2.2 Clustering Stability

One gains confidence in a clustering solution if it is stable to minor perturbations of the data. Very generally, stability can be measured by perturbing the data, with bootstrap or added noise, clustering the perturbed data, computing the pairwise distance between the original clusters and the new perturbed clusters, and normalizing the distances to get a measure of stability. Normalization is most commonly an average of the pairwise distances, but there are other methods [4, 16, 47].

Recently, the authors in [49] proposed a tightness measure (valued between 0 and 1) to reflect the stability of each cluster and an average tightness for measuring the overall stability of a partition. The key idea behind this stability measure is to determine a covering point set (CPS) for each cluster. There is a “match” relationship between cluster $C_i^{(1)}$ and $C_j^{(2)}$ from two partitions $\mathcal{P}^{(1)}$ and $\mathcal{P}^{(2)}$, if roughly speaking, the two clusters are the same despite possibly distinct labels. Suppose for a cluster S_k in the reference partition of the original data, there is a set of matched clusters $S_i, i = 1, \dots, m$, from partitions of perturbed datasets. The CPS S_α of cluster S_k at a coverage level α is defined as the smallest set such that at least $100(1 - \alpha)\%$ of the S_i clusters are subsets of S_α . In other words, the goal is to solve the optimization problem

$$S_\alpha = \arg \min_S |S|, \text{ s.t. } \sum_{i=1}^m 1_{(S_i \subset S)} \geq m(1 - \alpha).$$

The tightness of cluster S_k is defined as

$$R_t(k \mid S_\alpha) = \frac{\sum_{i=1}^m |S_i| / |S_\alpha|}{M},$$

where M is the total number of partitions and $|\cdot|$ denotes the cardinality of a set. Higher tightness values indicate higher stability.

We illustrate stability and tightness of two clusters found by the denoising algorithms DADA2 [9] (version 1.16.0) and AmpliCI [61] on the previously mentioned HIV data using the R package **OTclust** [49]. We randomly select 5% of the reads in the original data and mutate 5% of the sites with probability $\frac{1}{3}$ to one of the other nucleotides to generate five perturbed datasets. The average stability is 0.86 for DADA2 and 0.81 for AmpliCI. The lower stability of AmpliCI is caused by some 0 tightness clusters. Among the 32 clusters centered on the same ASVs recovered by both methods, about 60% of AmpliCI clusters have higher stability than the corresponding DADA2 clusters. Figure 4 shows the 90% CPS plot of two clusters from AmpliCI and DADA2 with matching ASVs and Fig. 5 shows the membership heat map of the same two clusters. Clusters with high stability should contain only and all high frequency points. The membership heat map and CPS plot together help us visualize the stability and uncertainty of a predicted cluster. Comparing cluster 1 for both methods, AmpliCI includes more distant members in the CPS, but DADA2 sometimes excludes core members of this cluster and includes, with high confidence by the membership heat map (Fig. 5), a small cluster, shown at the bottom right and far away from the main cluster. Based on the membership heat map of cluster 1, DADA2 is less stable than AmpliCI since some reads are not consistently included in cluster 1. In contrast, for the small cluster (cluster 39 of DADA2 and 28 of AmpliCI), DADA2 is more stable. Interestingly, t-SNE [81], which was used for the visualization, seems to suggest that cluster 1 includes some nearby satellite clusters. These satellite clusters may be amplified PCR errors or more troubling, true biological variants not included in the reference.

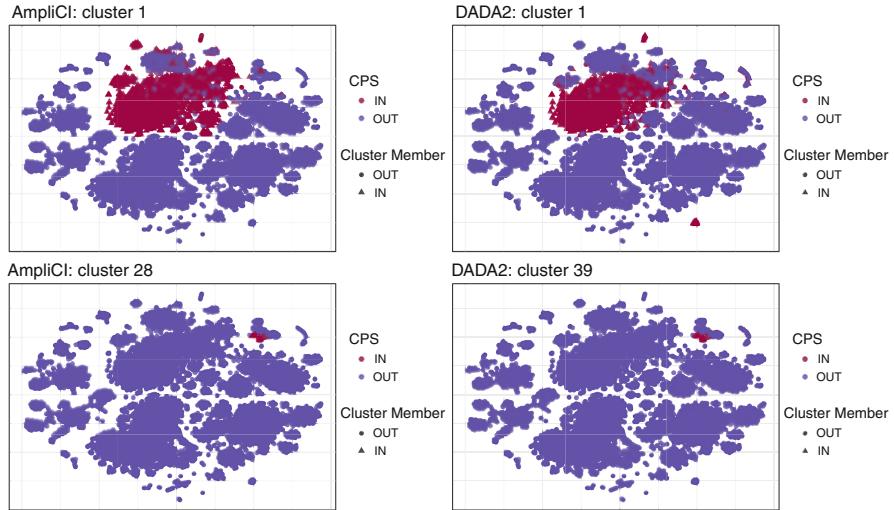


Fig. 4 t-SNE [81] visualization of Covering Point Sets (CPS) for HIV data. The 90% CPS plot of two clusters obtained from AmpliCI and DADA2 around identical ASVs. Red indicates a point, representing a read, is inside the CPS of the cluster

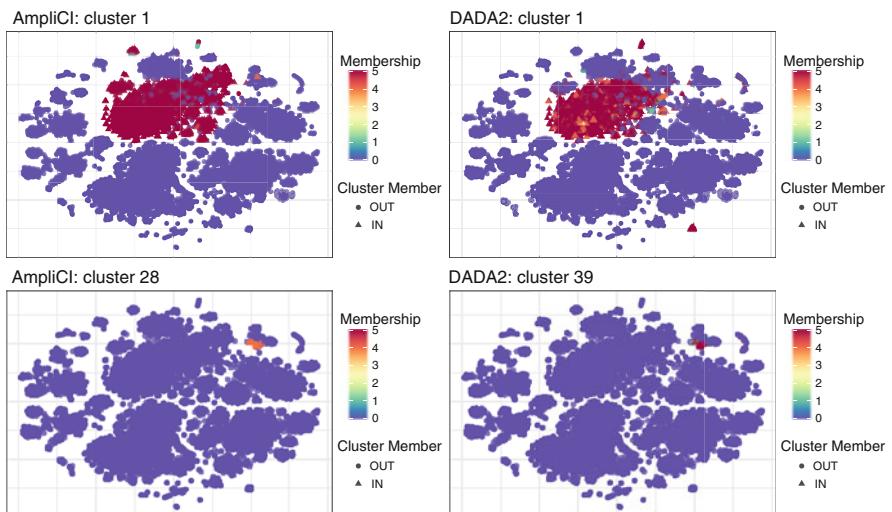


Fig. 5 t-SNE visualization of membership heat map for HIV data. Membership heat map of two clusters obtained from AmpliCI and DADA2. Different colors represent the number of times a read was assigned to the cluster across perturbed datasets. Triangles represent reads in a cluster; dots represent reads in the other clusters

5 Conclusions

The processing of amplicon sequence data from microbiome communities has greatly improved after a decade of progress. Technical improvements have reduced the many sources of bias, while statistical and bioinformatics techniques have improved the data processing. Modern denoising methods can now detect single-nucleotide variants in a mixed sample without relying on a reference database.

However, there remain persistent challenges facing both the technical aspects of data generation and the statistical data analysis. One challenge for denoisers is their current inability to detect PCR errors and amplification bias as well as other contamination products. These reads are high quality and look, in many ways, just like natural biological variation. It is disconcerting that among the three main denoising methods, DADA2, Deblur, and UNOISE3, and our own contribution AmpliCI, there is extensive disagreement except on the cleanest datasets. This disagreement strongly suggests that our understanding of noise in amplicon data is incomplete. Tools have been developed to remove PCR errors through a second round of sequence clustering [65] and contaminants via a *post hoc* statistical test [17]. One difficulty in assessing the methods is the lack of realistic amplicon read simulators or mock data without ambiguity. We believe that it will take clever protocols and supplemental information, such as that provided by unique molecular identifies (UMIs) or spike-in controls [87], to accurately compare the methods and point to potential improvements.

It is possible that shotgun sequencing methods or unamplified long-read technology will completely supplant amplicon-based methods because they avoid amplification bias, PCR errors, and amplification-induced contamination, but other biases, contaminants, and sequencing errors will persist and new challenges will emerge. There is much opportunity for development of methods in the emerging technologies and still improvements needed for amplicon sequencing. Certainly, the identification and quantification of microbiome communities will continue well into the future.

References

1. Amir, A., McDonald, D., Navas-Molina, J.A., Kopylova, E., Morton, J.T., Zech Xu, Z., Kightley, E.P., Thompson, L.R., Hyde, E.R., Gonzalez, A., Knight, R.: Deblur rapidly resolves single-nucleotide community sequence patterns. *mSystems* **2**(2), e00191–16 (2017). <https://doi.org/10.1128/mSystems.00191-16>
2. Apprill, A., McNally, S., Parsons, R., Weber, L.: Minor revision to V4 region SSU rRNA 806R gene primer greatly increases detection of SAR11 bacterioplankton. *Aquat. Microb. Ecol.* **75**(2), 129–137 (2015). <https://doi.org/10.3354/ame01753>
3. Ardui, S., Ameur, A., Vermeesch, J.R., Hestand, M.S.: Single molecule real-time (SMRT) sequencing comes of age: applications and utilities for medical diagnostics. *Nucleic Acids Res.* **46**(5), 2159–2168 (2018). <https://doi.org/10.1093/nar/gky066>

4. Bertoni, A., Valentini, G.: Model order selection for bio-molecular data clustering. *BMC Bioinformatics* **8**(Suppl 2), S7 (2007). <https://doi.org/10.1186/1471-2105-8-S2-S7>
5. Besser, J., Carleton, H.A., Gerner-Smidt, P., Lindsey, R.L., Trees, E.: Next-generation sequencing technologies and their application to the study and control of bacterial infections. *Clin. Microbiol. Infect.* **24**(4), 335–341 (2018). <https://doi.org/10.1016/j.cmi.2017.10.013>
6. Best, K., Oakes, T., Heather, J.M., Shawe-Taylor, J., Chain, B.: Computational analysis of stochastic heterogeneity in PCR amplification efficiency revealed by single molecule barcoding. *Scientific Reports* **5**(11), 14629 (2015). <https://doi.org/10.1038/srep14629>
7. Bolyen, E., Rideout, J.R., Dillon, M.R., Bokulich, N.A., Abnet, C.C., Al-Ghalith, G.A., Alexander, H., Alm, E.J., Arumugam, M., Asnicar, F., Bai, Y., Bisanz, J.E., Bittinger, K., Brejnrod, A., Brislawn, C.J., Brown, C.T., Callahan, B.J., Caraballo-Rodríguez, A.M., Chase, J., Cope, E.K., Da Silva, R., Diener, C., Dorrestein, P.C., Douglas, G.M., Durall, D.M., Duvallet, C., Edwardson, C.F., Ernst, M., Estaki, M., Fouquier, J., Gauglitz, J.M., Gibbons, S.M., Gibson, D.L., Gonzalez, A., Gorlick, K., Guo, J., Hillmann, B., Holmes, S., Holste, H., Huttenhower, C., Huttley, G.A., Janssen, S., Jarmusch, A.K., Jiang, L., Kaehler, B.D., Kang, K.B., Keefe, C.R., Keim, P., Kelley, S.T., Knights, D., Koester, I., Kosciolak, T., Kreps, J., Langille, M.G.I., Lee, J., Ley, R., Liu, Y.X., Loftfield, E., Lozupone, C., Maher, M., Marotz, C., Martin, B.D., McDonald, D., McIver, L.J., Melnik, A.V., Metcalf, J.L., Morgan, S.C., Morton, J.T., Naimey, A.T., Navas-Molina, J.A., Nothias, L.F., Orchanian, S.B., Pearson, T., Peoples, S.L., Petras, D., Preuss, M.L., Pruesse, E., Rasmussen, L.B., Rivers, A., Robeson, M.S., Rosenthal, P., Segata, N., Shaffer, M., Shiffer, A., Sinha, R., Song, S.J., Spear, J.R., Swafford, A.D., Thompson, L.R., Torres, P.J., Trinh, P., Tripathi, A., Turnbaugh, P.J., Ul-Hasan, S., van der Hooft, J.J.J., Vargas, F., Vázquez-Baeza, Y., Vogtmann, E., von Hippel, M., Walters, W., Wan, Y., Wang, M., Warren, J., Weber, K.C., Williamson, C.H.D., Willis, A.D., Xu, Z.Z., Zaneveld, J.R., Zhang, Y., Zhu, Q., Knight, R., Caporaso, J.G.: Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nature Biotechnology* **37**(8), 852–857 (2019). <https://doi.org/10.1038/s41587-019-0209-9>
8. Callahan, B.J., McMurdie, P.J., Holmes, S.P.: Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *ISME J.* **11**(12), 2639–2643 (2017). <https://doi.org/10.1038/ismej.2017.119>
9. Callahan, B.J., McMurdie, P.J., Rosen, M.J., Han, A.W., Johnson, A.J.A., Holmes, S.P.: DADA2: High-resolution sample inference from Illumina amplicon data. *Nature Methods* **13**(7), 581–583 (2016). <https://doi.org/10.1038/nmeth.3869>
10. Callahan, B.J., Wong, J., Heiner, C., Oh, S., Theriot, C.M., Gulati, A.S., McGill, S.K., Dougherty, M.K.: High-throughput amplicon sequencing of the full-length 16S rRNA gene with single-nucleotide resolution. *Nucleic Acids Res.* **47**(18), e103–e103 (2019). <https://doi.org/10.1093/nar/gkz569>
11. Caporaso, J.G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F.D., Costello, E.K., Fierer, N., Peña, A.G., Goodrich, J.K., Gordon, J.I., Huttley, G.A., Kelley, S.T., Knights, D., Koenig, J.E., Ley, R.E., Lozupone, C.A., McDonald, D., Muegge, B.D., Pirrung, M., Reeder, J., Sevinsky, J.R., Turnbaugh, P.J., Walters, W.A., Widmann, J., Yatsunenko, T., Zaneveld, J., Knight, R.: QIIME allows analysis of high-throughput community sequencing data. *Nature Methods* **7**(5), 335–336 (2010). <https://doi.org/10.1038/nmeth.f.303>
12. Caporaso, J.G., Lauber, C.L., Walters, W.A., Berg-Lyons, D., Huntley, J., Fierer, N., Owens, S.M., Betley, J., Fraser, L., Bauer, M., Gormley, N., Gilbert, J.A., Smith, G., Knight, R.: Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *ISME J.* **6**(8), 1621–1624 (2012). <https://doi.org/10.1038/ismej.2012.8>
13. Chan, A.W.Y., Naphtali, J., Schellhorn, H.E.: High-throughput DNA sequencing technologies for water and wastewater analysis. *Science Progress* **102**(4), 351–376 (2019). <https://doi.org/10.1177/0036850419881855>
14. Clooney, A.G., Fouhy, F., Sleator, R.D., O’ Driscoll, A., Stanton, C., Cotter, P.D., Claesson, M.J.: Comparing apples and oranges?: Next generation sequencing and its impact on microbiome analysis. *PLOS ONE* **11**(2), e0148028 (2016). <https://doi.org/10.1371/journal.pone.0148028>

15. D'Amore, R., Ijaz, U.Z., Schirmer, M., Kenny, J.G., Gregory, R., Darby, A.C., Shakya, M., Podar, M., Quince, C., Hall, N.: A comprehensive benchmarking study of protocols and sequencing platforms for 16S rRNA community profiling. *BMC Genomics* **17**(1), 55 (2016). <https://doi.org/10.1186/s12864-015-2194-9>
16. Datta, S., Datta, S.: Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics* **19**(4), 459–466 (2003). <https://doi.org/10.1093/bioinformatics/btg025>
17. Davis, N.M., Proctor, D.M., Holmes, S.P., Relman, D.A., Callahan, B.J.: Simple statistical identification and removal of contaminant sequences in marker-gene and metagenomics data. *Microbiome* **6**(1), 226 (2018). <https://doi.org/10.1186/s40168-018-0605-2>
18. Dopheide, A., Xie, D., Buckley, T.R., Drummond, A.J., Newcomb, R.D.: Impacts of DNA extraction and PCR on DNA metabarcoding estimates of soil biodiversity. *Methods Ecol. Evol.* **10**(1), 120–133 (2019). <https://doi.org/10.1111/2041-210X.13086>
19. dos Santos, H.R.M., Argolo, C.S., Argôlo-Filho, R.C., Loguercio, L.L.: A 16S rDNA PCR-based theoretical to actual delta approach on culturable mock communities revealed severe losses of diversity information. *BMC Microbiology* **19**(1), 74 (2019). <https://doi.org/10.1186/s12866-019-1446-2>
20. Edgar, R.: UNOISE3 command. https://www.drive5.com/usearch/manual/cmd_uneise3.html
21. Edgar, R.C.: Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* **26**(19), 2460–2461 (2010). <https://doi.org/10.1093/bioinformatics/btq461>
22. Edgar, R.C.: UPARSE: Highly accurate OTU sequences from microbial amplicon reads. *Nature Methods* **10**(10), 996–998 (2013). <https://doi.org/10.1038/nmeth.2604>
23. Edgar, R.C.: UNOISE2: improved error-correction for Illumina 16S and ITS amplicon sequencing. *bioRxiv* (2016). <https://doi.org/10.1101/081257>
24. Edgar, R.: Taxonomy annotation and guide tree errors in 16S rRNA databases. *PeerJ* **6**, e5030 (2018). <https://doi.org/10.7717/peerj.5030>
25. Eisenstein, M.: Microbiology: making the best of PCR bias. *Nature Methods* **15**(5), 317–320 (2018). <https://doi.org/10.1038/nmeth.4683>
26. Eloe-Fadrosh, E.A., Ivanova, N.N., Woyke, T., Kyrpides, N.C.: Metagenomics uncovers gaps in amplicon-based detection of microbial diversity. *Nature Microbiology* **1**(4), 15032 (2016). <https://doi.org/10.1038/nmicrobiol.2015.32>
27. Eren, A.M., Morrison, H.G., Lescault, P.J., Reveillaud, J., Vineis, J.H., Sogin, M.L.: Minimum entropy decomposition: Unsupervised oligotyping for sensitive partitioning of high-throughput marker gene sequences. *ISME J.* **9**(4), 968–979 (2015). <https://doi.org/10.1038/ismej.2014.195>
28. Ewing, B., Green, P.: Base-calling of automated sequencer traces using PHRED. II. error probabilities. *Genome Research* **8**(3), 186–194 (1998). <https://doi.org/10.1101/gr.8.3.186>
29. Ferretti, P., Farina, S., Cristofolini, M., Girolomoni, G., Tett, A., Segata, N.: Experimental metagenomics and ribosomal profiling of the human skin microbiome. *Experimental Dermatology* **26**(3), 211–219 (2017). <https://doi.org/10.1111/exd.13210>
30. Fouhy, F., Clooney, A.G., Stanton, C., Claesson, M.J., Cotter, P.D.: 16S rRNA gene sequencing of mock microbial populations- impact of DNA extraction method, primer choice and sequencing platform. *BMC Microbiology* **16**(1), 123 (2016). <https://doi.org/10.1186/s12866-016-0738-z>
31. Gohl, D.M., Vangay, P., Garbe, J., MacLean, A., Hauge, A., Becker, A., Gould, T.J., Clayton, J.B., Johnson, T.J., Hunter, R., Knights, D., Beckman, K.B.: Systematic improvement of amplicon marker gene methods for increased accuracy in microbiome studies. *Nature Biotechnology* **34**(9), 942–949 (2016). <https://doi.org/10.1038/nbt.3601>
32. Gołębiewski, M., Tretyn, A.: Generating amplicon reads for microbial community assessment with next-generation sequencing. *J. Appl. Microbiol.* **128**(2), 330–354 (2019). <https://doi.org/10.1111/jam.14380>
33. Hathaway, N.J., Parobek, C.M., Juliano, J.J., Bailey, J.A.: SeekDeep: single-base resolution de novo clustering for amplicon deep sequencing. *Nucleic Acids Res.* **46**(4), e21–e21 (2017). <https://doi.org/10.1093/nar/gkx1201>

34. Hebert, P.D.N., Braukmann, T.W.A., Prosser, S.W.J., Ratnasingham, S., DeWaard, J.R., Ivanova, N.V., Janzen, D.H., Hallwachs, W., Naik, S., Sones, J.E., Zakharov, E.V.: A Sequel to Sanger: amplicon sequencing that scales. *BMC Genomics* **19**(1), 219 (2018). <https://doi.org/10.1186/s12864-018-4611-3>
35. Hubert, L., Arabie, P.: Comparing partitions. *J. Classif.* **2**(1), 193–218 (1985). <https://doi.org/10.1007/BF01908075>
36. Jabara, C.B., Jones, C.D., Roach, J., Anderson, J.A., Swanstrom, R.: Accurate sampling and deep sequencing of the HIV-1 protease gene using a Primer ID. *Proc. Natl. Acad. Sci.* **108**(50), 20166–20171 (2011). <https://doi.org/10.1073/pnas.1110064108>
37. Janda, J.M., Abbott, S.L.: 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: Pluses, perils, and pitfalls. *J. Clin. Microbiol.* **45**(9), 2761 (2007). <https://doi.org/10.1128/JCM.01228-07>
38. Karst, S.M., Ziels, R.M., Kirkegaard, R.H., Sørensen, E.A., McDonald, D., Zhu, Q., Knight, R., Albertsen, M.: Enabling high-accuracy long-read amplicon sequences using unique molecular identifiers with Nanopore or PacBio sequencing. *bioRxiv* (2020). <https://doi.org/10.1101/645903>
39. Katoh, K., Standley, D.M.: MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.* **30**(4), 772–780 (2013). <https://doi.org/10.1093/molbev/mst010>
40. Kennedy, K., Hall, M.W., Lynch, M.D.J., Moreno-Hagelsieb, G., Neufeld, J.D.: Evaluating bias of Illumina-based bacterial 16S rRNA gene profiles. *Appl. Environ. Microbiol.* **80**(18), 5717 (2014). <https://doi.org/10.1128/AEM.01451-14>
41. Kim, D., Hofstaedter, C.E., Zhao, C., Mattei, L., Tanes, C., Clarke, E., Lauder, A., Sherrill-Mix, S., Chehoud, C., Kelsen, J., Conrad, M., Collman, R.G., Baldassano, R., Bushman, F.D., Bittinger, K.: Optimizing methods and dodging pitfalls in microbiome research. *Microbiome* **5**(1), 52 (2017). <https://doi.org/10.1186/s40168-017-0267-5>
42. Kinde, I., Wu, J., Papadopoulos, N., Kinzler, K.W., Vogelstein, B.: Detection and quantification of rare mutations with massively parallel sequencing. *Proc. Natl. Acad. Sci.* **108**(23), 9530–9535 (2011). <https://doi.org/10.1073/pnas.1105422108>
43. Knight, R., Vrbanac, A., Taylor, B.C., Aksенov, A., Callewaert, C., Debelius, J., Gonzalez, A., Kosciolet, T., McCall, L.I., McDonald, D., Melnik, A.V., Morton, J.T., Navas, J., Quinn, R.A., Sanders, J.G., Swafford, A.D., Thompson, L.R., Tripathi, A., Xu, Z.Z., Zaneveld, J.R., Zhu, Q., Caporaso, J.G., Dorrestein, P.C.: Best practices for analysing microbiomes. *Nat. Rev. Microbiol.* **16**(7), 410–422 (2018). <https://doi.org/10.1038/s41579-018-0029-9>
44. Konstantinidis, K.T., Tiedje, J.M.: Genomic insights that advance the species definition for prokaryotes. *Proc. Natl. Acad. Sci.* **102**(7), 2567–2572 (2005). <https://doi.org/10.1073/pnas.0409727102>
45. Kozlov, A.M., Zhang, J., Yilmaz, P., Glöckner, F.O., Stamatakis, A.: Phylogeny-aware identification and correction of taxonomically mislabeled sequences. *Nucleic Acids Res.* **44**(11), 5022–5033 (2016). <https://doi.org/10.1093/nar/gkw396>
46. Kumar, V., Vollbrecht, T., Chernyshev, M., Mohan, S., Hanst, B., Bavafa, N., Lorenzo, A., Kumar, N., Ketteringham, R., Eren, K., Golden, M., Oliveira, M.F., Murrell, B.: Long-read amplicon denoising. *Nucleic Acids Res.* **47**(18), e104–e104 (2019). <https://doi.org/10.1093/nar/gkz657>
47. Lange, T., Roth, V., Braun, M.L., Buhmann, J.M.: Stability-based validation of clustering solutions. *Neural Computation* **16**(6), 1299–1323 (2004). <https://doi.org/10.1162/089976604773717621>
48. Laursen, M.F., Dalgaard, M.D., Bahl, M.I.: Genomic GC-content affects the accuracy of 16S rRNA gene sequencing based microbial profiling due to PCR bias. *Front. Microbiol.* **8**, 1934 (2017). <https://doi.org/10.3389/fmicb.2017.01934>
49. Li, J., Seo, B., Lin, L.: Optimal transport, mean partition, and uncertainty assessment in cluster analysis. *Stat. Anal. Data Min.* **12**(5), 359–377 (2019). <https://doi.org/10.1002/sam.11418>

50. Lombard, N., Prestat, E., van Elsas, J.D., Simonet, P.: Soil-specific limitations for access and analysis of soil microbial communities by metagenomics. *FEMS Microbiol. Ecol.* **78**(1), 31–49 (2011). <https://doi.org/10.1111/j.1574-6941.2011.01140.x>
51. Malone, M., Gosbell, I.B., Dickson, H.G., Vickery, K., Espedido, B.A., Jensen, S.O.: Can molecular DNA-based techniques unravel the truth about diabetic foot infections? *Diabetes Metab. Res. Rev.* **33**(1), e2834 (2017). <https://doi.org/10.1002/dmrr.2834>
52. Mancabelli, L., Milani, C., Lugli, G.A., Fontana, F., Turroni, F., van Sinderen, D., Ventura, M.: The impact of primer design on amplicon-based metagenomic profiling accuracy: Detailed insights into Bifidobacterial community structure. *Microorganisms* **8**(1), 131 (2020). <https://doi.org/10.3390/microorganisms8010131>
53. Mignard, S., Flandrois, J.P.: 16S rRNA sequencing in routine bacterial identification: A 30-month experiment. *J. Microbiol. Methods* **67**(3), 574–581 (2006). <https://doi.org/10.1016/j.mimet.2006.05.009>
54. Müller, T., Ruppel, S.: Progress in cultivation-independent phyllosphere microbiology. *FEMS Microbiol. Ecol.* **87**(1), 2–17 (2014). <https://doi.org/10.1111/1574-6941.12198>
55. Murtagh, F.: A survey of recent advances in hierarchical clustering algorithms. *Comput. J.* **26**(4), 354–359 (1983). <https://doi.org/10.1093/comjnl/26.4.354>
56. Navarro, G.: A guided tour to approximate string matching. *ACM Comput. Surv.* **33**(1), 31–88 (2001). <https://doi.org/10.1145/375360.375365>
57. Navas-Molina, J.A., Peralta-Sánchez, J.M., González, A., McMurdie, P.J., Vázquez-Baeza, Y., Xu, Z., Ursell, L.K., Lauber, C., Zhou, H., Song, S.J., Huntley, J., Ackermann, G.L., Berg-Lyons, D., Holmes, S., Caporaso, J.G., Knight, R.: Chapter nineteen – advancing our understanding of the human microbiome using QIIME. In: DeLong, E.F. (ed.) *Microbial Metagenomics, Metatranscriptomics, and Metaproteomics, Methods in Enzymology*, vol. 531, pp. 371–444 (2013). <https://doi.org/10.1016/B978-0-12-407863-5.00019-8>
58. Nearing, J.T., Douglas, G.M., Comeau, A.M., Langille, M.G.I., Chen, J.: Denoising the denoisers: an independent evaluation of microbiome sequence error-correction approaches. *PeerJ* **6**, e5364 (2018). <https://doi.org/10.7717/peerj.5364>
59. Panek, M., Čipčić Paljetak, H., Barešić, A., Perić, M., Matijašić, M., Lojkic, I., Vranešić Bender, D., Krznarić, Ž., Verbanac, D.: Methodology challenges in studying human gut microbiota – effects of collection, storage, DNA extraction and next generation sequencing technologies. *Scientific Reports* **8**(1), 5143 (2018). <https://doi.org/10.1038/s41598-018-23296-4>
60. Parada, A.E., Needham, D.M., Fuhrman, J.A.: Every base matters: assessing small subunit rRNA primers for marine microbiomes with mock communities, time series and global field samples. *Environmental Microbiology* **18**(5), 1403–1414 (2016). <https://doi.org/10.1111/1462-2920.13023>
61. Peng, X., Dorman, K.: AmplicI: A high-resolution model-based approach for denoising Illumina amplicon data. *Bioinformatics* (btaa648) (2020). <https://doi.org/10.1093/bioinformatics/btaa648>
62. Posada-Cespedes, S., Seifert, D., Beerenwinkel, N.: Recent advances in inferring viral diversity from high-throughput sequencing data. *Virus Research* **239**, 17–32 (2017). <https://doi.org/10.1016/j.virusres.2016.09.016>
63. Prodan, A., Tremaroli, V., Brolin, H., Zwinderman, A.H., Nieuwdorp, M., Levin, E.: Comparing bioinformatic pipelines for microbial 16S rRNA amplicon sequencing. *PLOS ONE* **15**(1), e0227434 (2020). <https://doi.org/10.1371/journal.pone.0227434>
64. Quince, C., Lanzén, A., Curtis, T.P., Davenport, R.J., Hall, N., Head, I.M., Read, L.F., Sloan, W.T.: Accurate determination of microbial diversity from 454 pyrosequencing data. *Nature Methods* **6**(9), 639–641 (2009). <https://doi.org/10.1038/nmeth.1361>
65. Quince, C., Lanzén, A., Davenport, R.J., Turnbaugh, P.J.: Removing noise from pyrosequenced amplicons. *BMC Bioinformatics* **12**(1), 38 (2011). <https://doi.org/10.1186/1471-2105-12-38>
66. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971). <https://doi.org/10.1080/01621459.1971.10482356>
67. Rhoads, A., Au, K.F.: PacBio sequencing and its applications. *Genomics Proteomics Bioinformatics* **13**(5), 278–289 (2015). <https://doi.org/10.1016/j.gpb.2015.08.002>

68. Ritari, J., Salojärvi, J., Lahti, L., de Vos, W.M.: Improved taxonomic assignment of human intestinal 16S rRNA sequences by a dedicated reference database. *BMC Genomics* **16**(1), 1–10 (2015). <https://doi.org/10.1186/s12864-015-2265-y>
69. Rogers, G.B.: The human microbiome: opportunities and challenges for clinical care. *Intern. Med. J.* **45**(9), 889–898 (2015). <https://doi.org/10.1111/imj.12650>
70. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 410–420. Association for Computational Linguistics, Prague, Czech Republic (2007)
71. Salonen, A., Nikkilä, J., Jalanka-Tuovinen, J., Immonen, O., Rajilić-Stojanović, M., Kekkonen, R.A., Palva, A., de Vos, W.M.: Comparative analysis of fecal DNA extraction methods with phylogenetic microarray: Effective recovery of bacterial and archaeal DNA using mechanical cell lysis. *J. Microbiol. Methods* **81**(2), 127–134 (2010). <https://doi.org/10.1016/j.mimet.2010.02.007>
72. Salter, S.J., Cox, M.J., Turek, E.M., Calus, S.T., Cookson, W.O., Moffatt, M.F., Turner, P., Parkhill, J., Loman, N.J., Walker, A.W.: Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. *BMC Biology* **12**(1), 87 (2014). <https://doi.org/10.1186/s12915-014-0087-z>
73. Schirmer, M., D'Amore, R., Ijaz, U.Z., Hall, N., Quince, C.: Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data. *BMC Bioinformatics* **17**(1), 125 (2016). <https://doi.org/10.1186/s12859-016-0976-y>
74. Schloss, P.D., Westcott, S.L., Ryabin, T., Hall, J.R., Hartmann, M., Hollister, E.B., Lesniewski, R.A., Oakley, B.B., Parks, D.H., Robinson, C.J., Sahl, J.W., Stres, B., Thallinger, G.G., Van Horn, D.J., Weber, C.F.: Introducing Mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.* **75**(23), 7537–7541 (2009). <https://doi.org/10.1128/AEM.01541-09>
75. Sinha, R., Abu-Ali, G., Vogtmann, E., Fodor, A.A., Ren, B., Amir, A., Schwager, E., Crabtree, J., Ma, S., Abnet, C.C., Knight, R., White, O., Huttenhower, C., The Microbiome Quality Control Project Consortium: Assessment of variation in microbial community amplicon sequencing by the Microbiome Quality Control (MBQC) project consortium. *Nature Biotechnology* **35**(11), 1077–1086 (2017). <https://doi.org/10.1038/nbt.3981>
76. Stackebrandt, E., Goebel, B.M.: Taxonomic note: A place for DNA-DNA reassociation and 16S rRNA sequence analysis in the present species definition in bacteriology. *Int. J. Syst. Evol. Microbiol.* **44**(4), 846–849 (1994). <https://doi.org/10.1099/00207713-44-4-846>
77. Sze, M.A., Schloss, P.D.: The impact of DNA polymerase and number of rounds of amplification in PCR on 16S rRNA gene sequence data. *mSphere* **4**(3), e00163–19 (2019). <https://doi.org/10.1128/mSphere.00163-19>
78. Tikhonov, M., Leach, R.W., Wingreen, N.S.: Interpreting 16S metagenomic data without clustering to achieve sub-OTU resolution. *ISME J.* **9**(1), 68–80 (2015). <https://doi.org/10.1038/ismej.2014.117>
79. Tremblay, J., Singh, K., Fern, A., Kirton, E.S., He, S., Woyke, T., Lee, J., Chen, F., Dangl, J.L., Tringe, S.G.: Primer and platform effects on 16S rRNA tag sequencing. *Front. Microbiol.* **6**, 771 (2015). <https://doi.org/10.3389/fmicb.2015.00771>
80. Tremlett, H., Bauer, K.C., Appel-Cresswell, S., Finlay, B.B., Waubant, E.: The gut microbiome in human neurological disease: A review. *Ann. Neurol.* **81**(3), 369–382 (2017). <https://doi.org/10.1002/ana.24901>
81. van der Maaten, L., Hinton, G.: Visualizing high-dimensional data using t-SNE. *J. Mach. Learn. Res.* **9**(86), 2579–2605 (2008)
82. Vos, M., Wolf, A.B., Jennings, S.J., Kowalchuk, G.A.: Micro-scale determinants of bacterial diversity in soil. *FEMS Microbiol. Rev.* **37**(6), 936–954 (2013). <https://doi.org/10.1111/1574-6976.12023>

83. Wen, C., Wu, L., Qin, Y., Van Nostrand, J.D., Ning, D., Sun, B., Xue, K., Liu, F., Deng, Y., Liang, Y., Zhou, J.: Evaluation of the reproducibility of amplicon sequencing with Illumina MiSeq platform. *PLOS ONE* **12**(4), e0176716 (2017). <https://doi.org/10.1371/journal.pone.0176716>
84. Westcott, S.L., Schloss, P.D.: De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. *PeerJ* **3**, e1487 (2015). <https://doi.org/10.7717/peerj.1487>
85. Wick, R.R., Judd, L.M., Holt, K.E.: Deepbinner: Demultiplexing barcoded Oxford Nanopore reads with deep convolutional neural networks. *PLOS Comput. Biol.* **14**(11), 1–11 (2018). <https://doi.org/10.1371/journal.pcbi.1006583>
86. Yang, B., Wang, Y., Qian, P.Y.: Sensitivity and correlation of hypervariable regions in 16S rRNA genes in phylogenetic analysis. *BMC Bioinformatics* **17**(1), 135 (2016). <https://doi.org/10.1186/s12859-016-0992-y>
87. Zinter, M.S., Mayday, M.Y., Ryckman, K.K., Jelliffe-Pawlowski, L.L., DeRisi, J.L.: Towards precision quantification of contamination in metagenomic sequencing experiments. *Microbiome* **7**(1), 62 (2019). <https://doi.org/10.1186/s40168-019-0678-6>

Statistical and Computational Methods for Analysis of Shotgun Metagenomics Sequencing Data



Hongzhe Li and Haotian Zheng

1 Introduction

Microbiome consists of all the microorganisms in and on human body. These microbes play important roles in human health and disease. High-throughput shotgun metagenomic sequencing approaches enable genomic analyses of all microbes in a sample, not just those that are amenable to cultivation. In a typical metagenomic sequencing study, an average of 10 million reads are often obtained for a given sample. Such shotgun sequencing reads can be used to profile taxonomic composition and functional potential of microbial communities and to recover whole-genome sequences. Due to complexity and large volume of the data, analysis of shotgun sequencing reads data is more challenging than the marker-gene-based sequencing such as 16S rRNA sequencing in microbiome studies (Quince et al. [28]).

Metagenomic sequencing has wide applications in many areas of biomedical research, including microbiome and disease association studies, diagnosis and treatment of infection diseases, and studies of human host gene expressions and antimicrobial resistance. Depending on the studies and goals, different important microbial features can be derived from shotgun metagenomic data. For example, in disease association studies, useful features can be species abundance, metagenome single-nucleotide polymorphisms (SNPs), metagenome structural variants, and bacterial growth rates. In studies that integrate microbiome and host metabolome, useful features can be collection of all the biosynthetic gene clusters (BGCs).

H. Li (✉) · H. Zheng

Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, Philadelphia, PA, USA

e-mail: hongzhe@upenn.edu; Haotian.Zheng@upenn.edu

In infectious disease and antimicrobial resistance research, one is interested in identifying new bacterial species or strains that lead to the infectious disease.

The main computational problems in analysis of such shotgun short read data include: (1) binning problem that assigns taxonomic labels to these short DNA reads using sequencing alignment or machine learning methods; (2) quantifying the relative abundances of species, genes, or pathways; (3) metagenomic sequencing assemblies to discover new species; (4) strain-level analysis; and (5) estimation of metabolomic potentials. These computational problems are big data problems that involve merging hundreds of millions of shot sequencing reads with close to 282,580 complete genome sequences of prokaryotes (<https://www.ncbi.nlm.nih.gov/genome/browse#!/prokaryotes/>). Breitwieser et al. [3] reviewed the methods and databases for metagenomic classification and assembly. Most of the efficient computational tools and software packages have been developed by computational biologists and computer scientists. In this chapter, we summarize and review some of the most commonly used algorithms in the field of microbiome and metagenomic data analysis, focusing on the statistical and computational aspects of the methods, and also point out possible improvements and areas that require further research.

2 Methods for Species Identification and Quantification of Microorganisms

One basic feature of a microbial community is the relative abundance of different species in the community. Given short reads data from shotgun metagenomic sequencing, the first step of analysis is to identify and quantify the relative abundances of all the species in the study samples. This can be achieved by aligning the sequencing reads to the reference genomes. Many computational methods have been developed for taxonomic classification and quantification, see Ye et al. [34] for a benchmarking comparison of various methods in terms of accuracy and computing resources needed. One challenge is how to assign the ambiguous reads that originate from genomic locations shared among multiple groups of organisms.

There are two general approaches to tackle this challenge. The first approach is the marker-gene-based methods where marker genes with sequences that are unique to a clade are identified and reads are only aligned to these marker genes. This method represents taxonomic clades uniquely by sequences that do not share common regions with other clades of the same taxonomic rank. The marker genes can be clade-specific as used in *MetaPhlAn2* (Truong et al. [31]) or universal marker genes as used in *MOTU* (Sunagawa et al. [30]). By aligning reads only to these clade-specific marker genes, the problem of aligning ambiguous reads is solved. *MetaPhlAn2* pipeline has been used in the Human Microbiome Project and the Integrative Human Microbiome Project and is very widely used. *MetaPhlAn2* outputs the taxonomic relative abundance estimation at various taxonomic levels.

The second approach is based on using the full set of reference sequences available as a database and assigning ambiguous reads to their least common ancestor (LCA) in a taxonomic tree. *Kraken* (Wood and Salzberg [33]), a k -mer-based read binning method, is an example of such an approach. *Kraken* uses a database comprising a hash table of k -mers (k is about 31 and should be large) and their corresponding node in a given taxonomic tree. Then, it assigns reads based on where the majority of its k -mers are located in the tree. Whenever no clear vote by the k -mers of the read exists, *Kraken* assigns that read to its least common ancestor. See Fig. 1 for an illustration of the steps of *Kraken*. *Kraken* is a very fast read binning method, which is also often used for taxonomic profiling. After reads are assigned to the taxonomic tree, further processing is needed to estimate the relative abundance of the species in order to account for the uncertainty of the reads that are assigned to the LCA nodes. *Bracken* (Lu et al. [20]) addresses this problem by probabilistically re-assigning reads from intermediate taxonomic nodes to the species level or above.

The output from *Kraken* is read count at each node of the taxonomic tree, similar to read placement for 16s rRNA sequencing reads. One can apply the methods that take into account the taxonomic tree structure in microbiome data analysis. Wang, Cai and Li [32] presented a method that is based on flow on the tree, which can be extended for the data from *Kraken*.

It should be emphasized that shotgun metagenomic sequencing data only provides information on the relative abundance of the species in the community. Such data are compositional and require special care in their analysis (see Li [19] for a review of methods for analysis of microbiome compositional data).

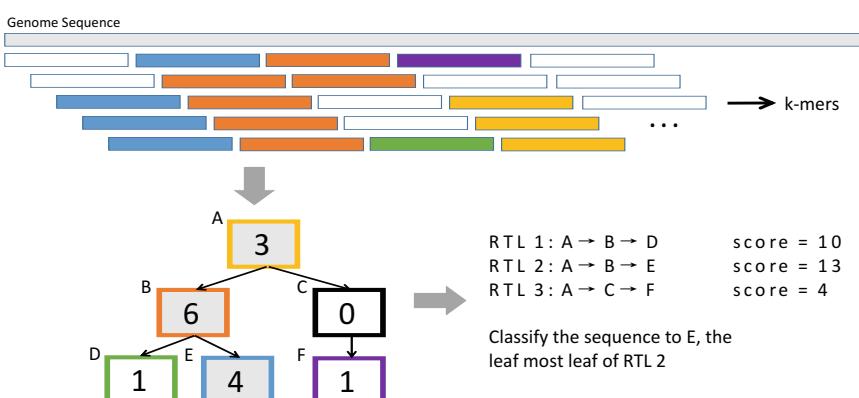


Fig. 1 Illustration of the *Kraken* algorithm for binning reads to taxon nodes on a taxonomic tree based on k -mer matching (modified from Figure 1 of Wood and Salzberg [33]). The number in each taxon node is the number of k -mers in the sequence that is associated with that taxon. The associated k -mers with each taxon node are marked with the corresponding color. The read sequence is assigned to the left-most leaf on the root-to-leaf (RTL) path with the greatest score, which is defined as the sum of the numbers in the nodes of the RTL path. The resulting tree can be used for taxonomic composition analysis and downstream statistical analysis

3 Metagenome Assembly and Applications

Besides quantifying the relative abundance of known bacterial species, new computational methods have also been developed for metagenome assemblies. The first step of metagenome assembly is to construct longer contiguous sequence based on the overlap of reads, a contig. These contigs are then clustered into bins based on their similarities. The algorithm outputs a large set of metagenome-assembled genomes (MAGs) (see Fig. 2 for an illustration), which are subject to downstream data analysis.

One important computational tool in genome assembly is to store the reads into the de Bruijn graph and to find Eulerian walks in the graph. Due to the large read counts for metagenomic data, metagenome assembly is time- and memory consuming. de Bruijn graph and Eulerian walks are powerful tools in computational genome sequence data analysis, but they are less known among statisticians. We briefly review the key concept in this section and point to the statistical questions.

3.1 *de Bruijn Assembly of a Single Genome*

de Bruijn graph, which is used widely in genome assembly, is a concept originated from graph theory. An n -dimensional de Bruijn graph of m symbols is basically a directed graph representing overlaps between sequences of symbols. It has m^n vertices, consisting of all possible length- n sequences of the given symbols. If one of the vertices can be expressed as another vertex by shifting all its symbols by one place to the left and adding a new symbol at the end of this vertex, then the latter has a directed edge to the former vertex. In genome assembly, it is explicit to create an assembly graph to illustrate the connecting relationships between reads or contigs. Oftentimes in an assembly graph, nodes represent DNA sequences (unitigs/contigs), while edges represent overlaps between those sequences. An assembly graph represents fundamental uncertainty in possible paths to go through the sequences.

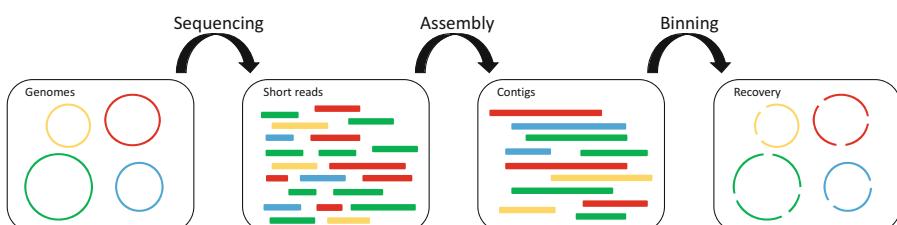


Fig. 2 Illustration of metagenome assembly to metagenome-assembled genomes (MAGs) that include a set of contigs. MAGHIT and MetaBAT2 are two most commonly used packages for assembly and for binning, respectively

de Bruijn graph can be used to construct an assembly graph based on the data of sequencing reads. The key point is to connect two substrings (represented by vertices) in a de Bruijn graph only if there is a read showing one substring can be transformed by shifting all its symbols by one place to the left and adding a new symbol at the end of this substring to another through that read. For instance, if there is a read whose sequence is GCCCA, as well as two substrings GCCC and CCCA, we can add an edge from the vertex representing GCCC to the vertex of CCCA. However, if there is not a read containing GCCCT as a part of it, even if there could be a vertex representing the substring CCCT, we should not add an edge from GCCC to CCCT in the de Bruijn graph. To make a de Bruijn graph consistent inside, we will need reads of length L , and they should overlap by $L-1$ bases. However, in most of the real cases, neither all reads overlap with each other perfectly, nor all reads have the same length. To resolve those problems, all k -length subsequences of the reads, i.e., the k -mers, are often used in genome assembly.

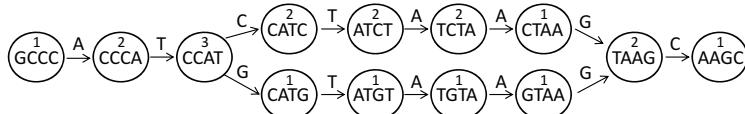
To construct a de Bruijn graph, we start from dividing each read into several k -mers with a pre-specified k . We traverse all of the k -mers of a given read and form the left $k - 1$ -mer (a substring with length $k - 1$) and the right $k - 1$ -mer of each k -mer. We include all the possible $k - 1$ -mers as vertex in the prospective de Bruijn graph and draw a directed edge from each left $k - 1$ -mer to its corresponding right $k - 1$ -mer. If the left and right $k - 1$ -mers are the same in a k -mer, we will draw an edge to itself. In the illustrative example in Fig. 3, we have three reads, CCCATGTAAG, CCATCTAAC, and GCCCATCTA. We set $k = 5$ and find all of the 5-mers of the reads. In the first read CCCATGTAAG, all the 5-mers are CCCAT, CCATG, CATGT, ATGTA, TGTA, and GTAAG. We then get the left and right 4-mers of each 5-mer and draw edges between them. There are edges from CCCA to CCAT, CCAT to CATG, CATG to ATGT, ATGT to TGTA, TGTA to GTAA, and GTAA to TAAG. We then construct a de Bruijn graph with all of the 4-mers of the 3 reads as vertices, which are shown in part (i) of Fig. 3, and draw an edge between two 4-mers if they together form a 5-mer of the reads. The constructed de Bruijn graph with the 3 reads above is shown in part (ii), where each vertex is a 4-mer, and the number in the vertex shows how many times that 4-mer appeared in all of the 3 reads. The letter on each edge indicated how a left 4-mer is transformed into its corresponding right 4-mer that is connected by that edge.

The next step in genome assembly is to find the origin genome sequence in the de Bruijn graph by looking for an Eulerian walk. If we manage to find an Eulerian walk in the de Bruijn graph, we then find the original genome sequence. After we build the de Bruijn graph as in Fig. 3, we next find a walk through it as a contig. In our example, one digit replacement, such as the C replaced by G in read 1, causes a branch of length 4 in the de Bruijn graph. In our example, we cannot find an Eulerian walk that visits each vertex exactly once, so we have to abandon a branch to get a walk through the graph. Here, we abandon the branch with lower frequency, which is defined as the sum of the numbers in the vertices on that branch, and choose the walk or branch with the highest frequency, shown in part (iii) of Fig. 3.

(i) Make k-mers

Read 1: CCCATGTAAG	Read 2: CCATCTAACG	Read 3: GCCCATCTA
k-mers: CCCA	CCAT	GCCC
CCAT	CATC	CCCA
CATG	ATCT	CCAT
ATGT	TCTA	CATC
TGTA	CTAA	ATCT
GTAA	TAAG	TCTA
TAAG	AAGC	

(ii) Build a De Bruijn Graph



(iii) Walk through the graph and find contigs

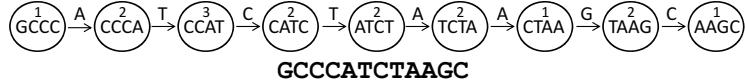


Fig. 3 An illustration of de Bruijn graph for genome assembly of three sequencing reads by using 4-mers, where nodes represent different k -mers and the numbers in the node indicate the number of the corresponding k -mer observed in the data. The Eulerian walk in the de Bruijn graph recovers the original genome sequence

3.2 Modification for Metagenome and Metagenome-Assembled Genomes

Various modifications of the methods for single-genome assembly have been made particularly for metagenome assembly to overcome the challenges of unknown abundance and diversity of the microbial community and related species in the metagenomes. Metagenome assembly graphs are frequently large, with millions of nodes, and require 10s to 100s of gigabytes of RAM for storage. Ayling et al. [2] present a review of various methods for metagenome assembly with short reads. Among various methods, *MEGAHIT* (Li et al. [18]) is most widely used method for contig construction. R package *bgtools* provides an interactive visualization tool for metagenomic bins, which is very useful for statisticians to explore the data (Seah and Gruber-Vodicka [29]).

MetaBAT2 (Kang et al. [15]) is most widely used computational package for binning the contigs. It performs pairwise comparisons of contigs by calculating probabilistic distances based on tetranucleotide frequency and then uses a k -medoid clustering algorithm to bin contigs to genomes.

Alternative to *MetaBAT2*, *CONCOCT* (Alneberg et al. [1]) is a binning method based on k -mer frequencies of the contigs. For metagenomics data, a co-assembly

of reads from all samples is first performed to obtain the set of contigs, which can be further filtered by length, and only contigs greater than a minimum size are used. For samples $j = 1, 2, \dots, M$, and contigs $i = 1, 2, \dots, N$, a coverage number Y_{ij} is defined as the average number of reads that are mapped to contig i per base from sample j . For each contig, we get a vector of coverage $Y_i = (Y_{i,1}, \dots, Y_{i,M})$ over M samples. In addition, a composition number is defined as the frequency for each k -mer and its reverse complement in that contig. For a fixed length k , the dimension of composition would be $V = f(k)$, which is the total number of possible k -mers, where reverse complements are considered as one possible k -mer. So for each contig i , we have its composition vector $Z_i = (Z_{i,1}, \dots, Z_{i,V})$, where $Z_{i,v}$ is the count of k -mer v that appeared in contig i . After adding pseudo-counts to remove zero in the input, together with normalization and logarithm transformation, a profile for contig i of dimension $E = M + V + 1$ is formed, where 1 comes from the total coverage for a contig in all the samples. *CONCOCT* performs a dimension reduction using principal-component analysis (PCA) and then clusters the contigs into bins using a Gaussian mixture model with a variational Bayesian approximation.

Using *MEGAHIT* and *MetaBAT2*, Pasolli et al. [27] leveraged 9,428 metagenomes to reconstruct 154,723 microbial genomes (45% of high quality) spanning body sites, ages, countries, and lifestyles. They recapitulated 4,930 species-level genome bins (SGBs), 77% without genomes in public repositories (unknown SGBs [uSGBs]). As microbial genomes are available at an ever-increasing pace, as cultivation and sequencing become cheaper, obtaining metagenome-assembled genomes (MAGs) becomes more effective. These unknown SGBs are expected to explain additional variability of the phenotypes of interest. Zhu et al. [35] showed that these reads from unknown organisms significantly increase the prediction accuracy of the disease status.

3.3 Compacted de Bruijn Graph

Shotgun metagenomic data also provide information on strain-level variation or metagenome structural variation. For strain-level analysis of metagenomes, compacted de Bruijn graph provides an efficient way of describing the data, where long simple paths of a de Bruijn graph are compacted into single vertices in order to reduce computational burden of the vast amount of k -mers. Here, the simple path to be compacted is also known as a unitig, which is defined as a path with all but the first vertex having in-degree 1, and all but the last vertex having out-degree 1. Here, the in-degree of a vertex is the number of edges pointing to that vertex in the de Bruijn graph, and the out-degree of a vertex is the number of edges pointing from that vertex. The graph after compaction is called a compact de Bruijn graph (cDBG). In a cDBG, one vertex may represent more than one k -mer, in contrast with one vertex representing one k -mer in a de Bruijn graph.

To illustrate the ideas, Fig. 4a shows a de Bruijn graph. In the path GGCC→GCC→CCCA, all vertices except for the first one, GGCC, have in-

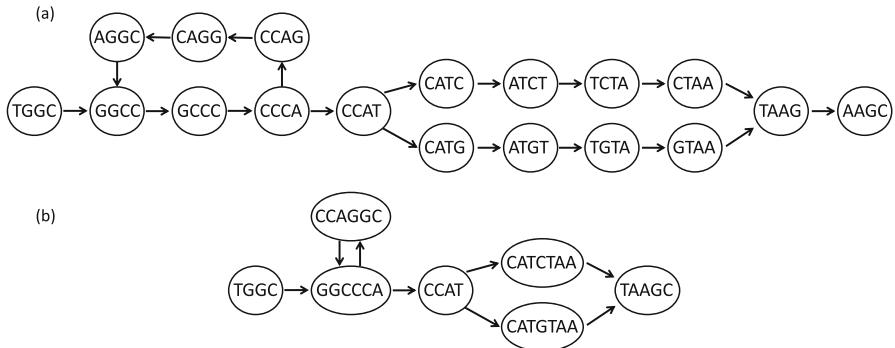


Fig. 4 Illustration of the compact de Bruijn graph (b) derived from the k -mer-based de Bruijn graph (a). For the de Bruijn graph, the nodes are k -mers, and for the compact de Bruijn graph, the nodes are contigs

degree equal to 1 (the in-degree of GGCC is 2), and all vertices except for the last one, CCCA, have out-degree 1. Therefore, the path GGCC → GCCC → CCCA is a “simple path” and can be compacted to GGCCCA. Similarly, in the paths CCAG → CAGG → AGGC, CATC → ATCT → TCTA → CTAA, and CATG → ATGT → TGTA → GTAA, all of their vertices have both in-degree and out-degree equal to 1, so they are simple paths and can be compacted to CCAGGC, CATCTAA, and CATGTAA, respectively. After we compacted all of the simple paths in the de Bruijn graph, we obtain the compact de Bruijn graph that is shown in Figure compact (b), where each vertex represents a unitig instead of one k -mer.

Chikhi et al. [6] developed an efficient algorithm BCALM2 to construct the cDBG. There are three main steps to get a compact de Bruijn graph from a set of k -mers its correspondingly formed de Bruijn graph operated from metagenome reads. The first step is to distribute the k -mers into buckets based on their “minimizers” (defined in Chikhi et al. [6]), with some k -mers being thrown into two buckets. Next, each bucket is compacted separately. Finally, the k -mers that were thrown into two buckets are glued back together so that duplicates are removed.

Using the compacted de Bruijn graph, Brown et al. [5] developed an efficient graph algorithm for investigating graph neighborhoods of a very large metagenome assembly de Bruijn graph. They developed and implemented a scalable graph query framework for extracting unbinned sequence from metagenome assembly graphs with millions of nodes by exploiting the structural sparsity of compact de Bruijn assembly graphs. These unbinned sequences can be further analyzed to discover new strains and new hidden sequence diversity. One application is to identify the genome neighborhood for a known bacterial genome. The reads from this neighborhood can be assembled and compared with the known genome to identify the strain variability of the known bacterium.

4 Estimation of Growth Rates for Metagenome-Assembled Genomes (MAGs)

The previous section reviews methods for metagenome assembly. In order to make the metagenomic data comparable across different samples, metagenome assembly has to be performed jointly over the combined reads of all the samples. After we obtain the contigs and bins, we usually align the metagenomic reads to each of the contigs to obtain the read coverage for each of the contigs and each of the samples. With appropriate normalization and correcting for possible GC bias, one can quantify the bacterial abundance based on these read coverage data.

Besides the relative abundance information, the uneven read coverage data can be used for estimating the bacterial growth dynamics or replication rates (Korem et al. [16]; Brown et al. [4]; Gao and Li [12]). Such bacterial replication rates provide important insights into the contribution of individual microbiome members to community functions. In a microbiome community, dividing cells are expected to contain, on average, more than one copy of their genome. Since the growing bacterial cells are unsynchronized and contain genomes that are replicated to different extents, we expect to observe a gradual reduction in the average genome copy number from the origin to the terminus of replication (Korem et al. [16]; Brown et al. [4]). This decrease in genome copy number can be detected by measuring changes in DNA sequencing coverage across complete genomes. Figure 5 illustrates this key idea. For the actively dividing bacteria, due to the bidirectional DNA replication from the replication starting sites, the read coverage is expected to decrease along the genome and the rate of decrease can be used to quantify the bacterial replication rate. Korem et al. [16] define the peak-to-trough ratio to quantify the bacterial replication rate for those bacteria with complete genome sequences available.

For MAGs, since we do not know the order of the contigs along the true genome, to estimate the replication rates, one has to first estimate the order of these contigs. Motivated by a simple linear growth model of DNAs, Gao and Li [12] proposed to apply PCA with contigs as observations to estimate the order, which has been shown to be very effective. Consider the following permuted monotone matrix model:

$$Y = \Theta\pi + Z, \quad (1)$$

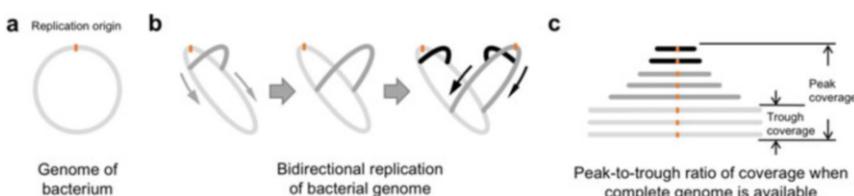


Fig. 5 Illustration of bacterial replication rate estimation. Bacterial circular genome (a), bidirectional replication (b), and peak-to-trough ratio of uneven read coverage (c)

where the observed data $Y \in \mathcal{R}^{n \times p}$ is the matrix of the preprocessed contig coverage for a given bacterial species. Specifically, the entry Y_{ij} represents the log-transformed averaged read counts of the j -th contig of the bacterial species for the i -th sample after the pre-processing steps, including genome assemblies, GC adjustment of read counts, and outlier filtering. In practice, the data set is usually high-dimensional in the sense that the number of contigs p far exceeds the sample size n . The signal matrix $\Theta \in \mathcal{R}^{n \times p}$ represents the true log-transformed coverage matrix of n samples and p contigs, where each row is monotone due to the bidirectional DNA replication mechanism. Under the permuted linear growth model, we assume that model (1) holds over the restricted set

$$\mathcal{D}_0 = \left\{ (\Theta, \pi) \in \mathcal{D} \times \mathcal{S}_p : \begin{array}{l} \theta_{ij} = a_i \eta_j + b_i, \text{ where } a_i, b_i \in \mathcal{R} \text{ for } 1 \leq i \leq n, \\ \eta_j \leq \eta_{j+1} \text{ for } 1 \leq j \leq p-1 \text{ and } \sum_{j=1}^p \eta_j = 0. \end{array} \right\}.$$

In other words, each row of Θ has a linear growth pattern with possibly different intercepts and slopes. Under this model, the true coverage matrix is rank-1. We consider the row-normalized observation matrix $X = Y(I_p - \frac{1}{p}ee^\top)$ and its first right singular vector, i.e.,

$$\hat{v} = (\hat{v}_1, \dots, \hat{v}_p)^\top = \arg \max_{v \in \mathcal{R}^p : \|v\|_2=1} v^\top X^\top X v.$$

Ma, Cai and Li [21] showed that the order statistics $\{\hat{v}_{(1)}, \dots, \hat{v}_{(p)}\}$ can be used to optimally recover the permutation π , or the original column orders, by tracing back the permutation map between the elements of \hat{v} and their order statistics.

As an example, Fig. 6a shows the read coverage for one MAG over its contigs for three gut microbiome samples with Crohn's disease from the study of Lewis et al. [17]. We cannot see any patterns of the data. However, after sorting the contigs based on the PCA, we observe a clear monotone pattern of the read coverage (see Fig. 6b). Based on this sorted contig coverage, Gao and Li [12] developed *DEMIC* to estimate the bacterial replication rates for the MAGs. Ma, Cai and Li [21] further showed that the PCA-based estimate of the ordered contigs achieves the minimax rate under certain conditions.

5 Methods for Identifying Biosynthetic Gene Clusters

The next phase of human microbiome research is moving from taxonomic and gene content profiling to functional microbiome by identifying, characterizing, and quantifying microbiome-derived small molecules that are responsible for a specific phenotype. Thousands of functionally interesting small molecules coded by various genes of microbiota have been discovered, including many antibiotics, toxins, pigments, immunosuppressants (Donia and Fischbach [9]). These small molecules

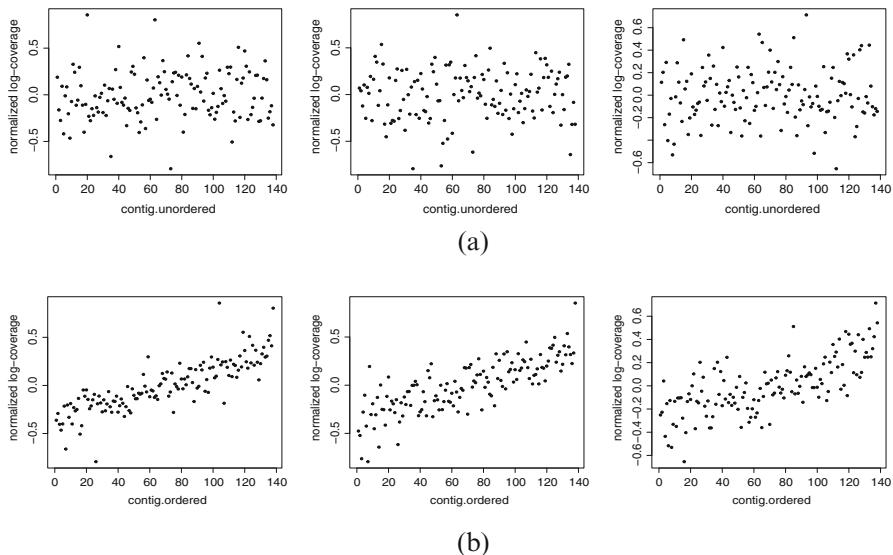


Fig. 6 Illustration of read coverage for an assembled genome for three gut samples of patients with Crohn's disease, where each dot represents a contig for the assembled genome. Y-axis: log of normalized read coverages. PCA is used to order the contigs based on the coverage over the samples. **(a)** Log read coverage for 3 children with Crohn's disease before contig ordering. **(b)** Log read coverage for 3 children with Crohn's disease after contig ordering

represent a major source of important nature products. Due to the wide range of bioactivities and pharmacological properties, identification of these natural products from microorganisms is an important problem in microbiome research.

The small molecules produced by bacteria are coded by biosynthetic gene clusters (BGCs) discovered along the bacterial genomes. These genes encode enzyme complexes or proteins participating in a common pathway that are continuously clustered in a chromosome region (see Fig. 7a). The BGCs are often collinearly arranged according to their biochemical reaction order (Cimermancic et al. [7]). The chemical and biological mechanisms of known BGCs such as non-ribosomal peptide synthetase (NRPS) and polyketide synthase (PKS) indicate that these multi-domain enzyme complexes are coordinated between the BGC genes. The end products of BGC pathways are bioactive small chemicals or nature products that are diverse in both structures and functions.

The Minimum Information about a Biosynthetic Gene Cluster (MIBiG) database (<https://mibig.secondarymetabolites.org>) includes an updated list of verified BGCs identified in various microorganisms and provides an important resource for BGC research (Medema et al. [24]). As an example, Fig. 7a shows the structure of BGC BGC0000007: aflatoxin biosynthetic gene cluster from *Aspergillus flavus*, which includes genes and their functions. New GBCs and their biosynthetic classes have been discovered and deposited into the database based on various experimental methods.

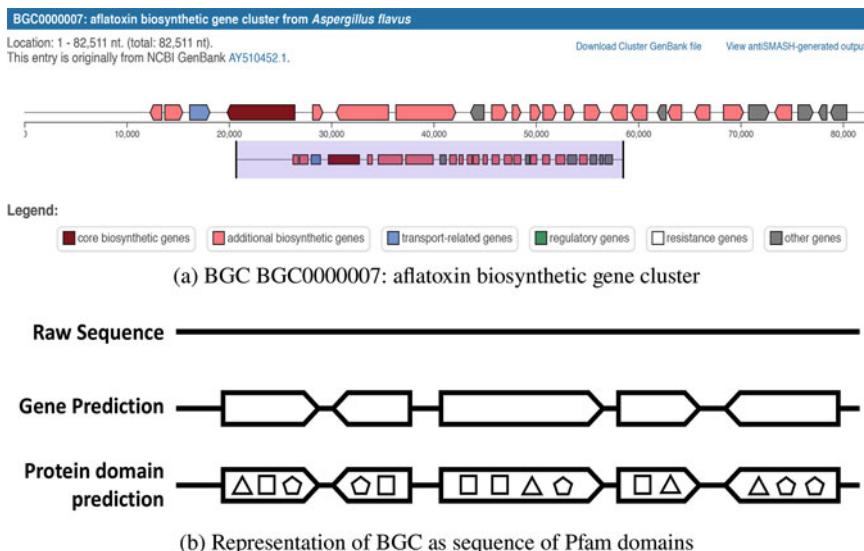


Fig. 7 (a) Illustration of BGC BGC0000007: aflatoxin biosynthetic gene cluster from *Aspergillus flavus*. <https://mibig.secondarymetabolites.org/repository/BGC0000007/index.html#r1c1>. (b) A BGC presented as a sequence of protein family (Pfam) domains (modified based on (Hannigan et al. [13]))

The BGCs listed under MIBiG are used in various computational methods for identifying new BGCs and predicting their classes, among which *ClusterFinder* and *DeepBGC* are the two state-of-the-art methods. Both *ClusterFinder* and *DeepBGC* are developed for identifying the BGCs in the bacteria with known complete genome sequences. *ClusterFinder* and *DeepBGC* use the Pfam domain sequential order information in BGC and non-BGC sequences in making the predictions. Specifically, raw genomic sequences are used for gene/ORF prediction using tools like *Prodigal* (Hyatt et al. [14]), and the Pfam domains are assigned to each ORF using *hmmscan* (Eddy [10]). Each BGC is then represented as a sequence of Pfam domains (see Fig. 7b for an illustration).

5.1 A Hidden Markov Model-Based Approach

Cimermancic et al. [7] developed a HMM probabilistic model (*ClusterFinder*), which provided a general solution for BGC identification for both well-studied and novel BGC classes. Using known gene annotations and predicted open reading frames (ORFs), *ClusterFinder* models the data at the protein family domain levels (Pfam) (Fig. 7b) and implements a standard two-stage HMM for estimating the posterior probability of being a BGC for each Pfam domain along the genome, where

the emission probabilities are simply the probabilities of observing a particular Pfam domain in BGCs and in non-BGC background. These probabilities are pre-estimated using the training data. HMM then estimates the posterior probability of being in BGC for each of the Pfam domain. The posterior probabilities are further processed to identify the BGCs.

Using *ClusterFinder*, they performed a systematic screening of BGCs in over 1000 bacterial genomes throughout the prokaryotic tree of life and revealed a striking finding of the predominance of Saccharides, a BGC class that has been overlooked in previous research. Compared to the traditional lab-based methods for BGC identification, their work shed light on the possibility of discovering unknown BGCs using computational methods, even for the less studied BGC classes.

5.2 A Deep Learning Approach

Following a similar setting as *ClusterFinder*, *DeepBGC* is the first attempt to employ nature language processing (NLP) and deep learning (DL) strategy for improved BGC identification (Hannigan et al. [13]), where the Pfam sequences of known BGCs and non-BGC are treated as labeled text data, with the Pfam names serving as words of the texts. As commonly used in DL and NLP, Word2Vec is used to learn word (Pfam domain names) embeddings with shallow two-layer neural network and outputs a set of numerical vectors. Word2Vec groups the vectors of similar Pfams together in vector space, where it detects similarities mathematically. Word2Vec creates vectors that are numerical representations of word features such as the context of individual Pfam. *ClusterFinder* then applies the bidirectional long short-term memory (BiLSTM) deep learning model to build predictive model for BGC vs. non-BGC. They showed *DeepBGC* outperformed *ClusterFinder* in both AUC and precision recall in detecting the BGCs on the same validation set. Unlike *ClusterFinder*, *DeepBGC* uses the Pfam domain sequential order information in BGC and non-BGC sequences in making the predictions. Specifically, each Pfam name is numerically coded using Pfam2vec trained using the Pfam names. The BiLSTM outputs classification score for each domain, and the domain scores are summarized across genes, which are selected accordingly as the BGCs. They showed improved performance of *DeepBGC* over the *ClusterFinder*.

5.3 BGC Identification Based on Metagenomic Data

Since both *ClusterFinder* and *DeepBGC* have limited their predictions of the BGCs in the bacteria with known complete genome sequences, with new metagenomic data being generated in very large scale, a logical next step is to identify possible new BGCs based on shotgun metagenomic data. Research in this direction is very limited.

One straightforward approach is to first perform metagenome assembly using the methods introduced in Sect. 3.2 and then apply methods such as *ClusterFinder* or *DeepBGS* to the genome assemblies. This approach was recently explored by Cuadrat et al. [8] to recover BGCs using metagenomic data sampled from Lake Stechlin. One limitation with this assembly-based method is that some BGCs might be scattered through multiple contigs, which make the direct application of *DeepBGC* or *ClusterFinder* infeasible, especially in the post-processing steps when the Pfam-specific predictions are combined into BGCs. Since the contigs in shotgun metagenomics are often short, the existing tools may fail to predict a large fraction of long BGCs.

Meleshko et al. [25] developed *biosyntheticSPAdes*, a tool for predicting BGCs in assembly graphs. This algorithm does not assume that each BGC is encoded within a single contig in the genome assembly, a condition that is violated for most sequenced microbial genomes where BGCs are often scattered through several contigs, making it difficult to reconstruct them. *biosyntheticSPAdes* involves identifying the Pfam domain edges in the assembly graph using HMMER (Eddy [11]), extracting BGC subgraphs, and restoring collapsed domain in the assembly graphs. This is another interesting application of the de Bruijn graph.

6 Future Directions

Shotgun metagenomics have an increasingly important part to play in diverse biomedical applications. We have reviewed some statistical and computational methods for analyzing the shotgun metagenomic data in microbiome studies, focusing more on the computational tools. We feel that it is important to understand how the raw sequencing reads data are processed to summarize the metagenomic data into biologically relevant features in order to understand the uncertainty and possible bias of such estimates. By using statistical inference ideas, we can improve some existing methods. For example, *DEMIC* (Gao and Li [12]) improves *iRep* (Brown et al. [4]) in estimating the bacterial replication rates by using the data across all samples in order to determine the contig order along the genome. Ma, Cai and Li [21] developed a permuted monotone matrix model and provided a theoretical justification of using the first right singular vector in ordering the contigs. They further showed that such a procedure is minimax rate optimal.

Although the methods we reviewed were largely developed by computational biologists or computer scientists, we think that statisticians should be more involved in these initial data processing steps as measurement determines downstream data analysis. When processing the raw sequencing data, we should be aware of the experimental bias, measurement errors, and possible batch effects. As an example, McLaren, Willis and Callahan [23] observed that the measured relative abundances within an experiment are biased by unknown but constant multiplicative factors. When bias acts consistently in this manner, it can be accounted for through the use

of bias-insensitive analyses such as ratio-based methods or corrected by a calibration procedure.

We can also make larger impact to metagenomic data analysis by further improving some of the methods based on either the intermediate or the final outputs from these efficient computational methods. For example, after we have the read placements on the taxonomic tree using *Kraken*, we may develop better statistical methods for quantifying the species abundance or identifying the bacterial taxa that are associated with outcomes. After we summarize the metagenomic data as k -mer counts using algorithm such as *JELLYFISH* (Marcais and Kingsfors [22]), we can develop methods to analyze such very large and potentially sparse count tables. Such alignment-free methods have recently been explored by Zhu et al. [35], who showed improvements in predicting diseases using the unaligned reads. Menegaux and Vert [26] proposed to bin together k -mers that appear together in the sequencing reads by learning a vector embedded for the vertices of a compacted de Bruijn graph, allowing us to embed any DNA sequence in a low-dimensional vector space where a machine learning system can be trained.

One challenge in analyzing metagenomic data is the volume of the data that requires large storage and computing power. Although great efforts have been devoted to improve the computation efficiency, for a typical metagenomic study of hundreds of subjects, it takes days to process the data using either *Kraken* or genome assembly. It is also very time-consuming to obtain the intermediate data such as counts of all 31-mers in a metagenomic sample used in *Kraken* algorithm or to construct the de Bruijn graph for shotgun data. Another challenge faced by statisticians is how to effectively access and utilize the data in the public domains, for example, all the BGCs and related information in the BGC repository (<https://mibig.secondarymetabolites.org/repository>) and the complete genome sequences of all the bacterial genomes.

Acknowledgments This work is supported by NIH grants GM123056 and GM129781.

References

1. Alneberg, J., Bjarnason, B., de Bruijn, I. et al.: Binning metagenomic contigs by coverage and composition. *Nature Methods* **11**, 1144–1146 (2014)
2. Ayling, M., Clark, M.D., Leggett, R.M.: New approaches for metagenome assembly with short reads. *Brief. Bioinform.* **21**(2), 584–594 (2020)
3. Breitwieser, F.P., Lu, J., Salzberg, S.L.: A review of methods and databases for metagenomic classification and assembly. *Brief. Bioinform.* **20**(4), 1125–1136 (2019)
4. Brown, C.T., Olm, M.R., Thomas, B.C., Banfield, J.F.: Measurement of bacterial replication rates in microbial communities. *Nature Biotechnology* **34**(12), 1256–1263 (2016)
5. Brown, C.T., Moritz, D., O'Brien, M.P., Reidl, F., Reiter, T., Sullivan, B.D.: Exploring neighborhoods in large metagenome assembly graphs using spacegraphcats reveals hidden sequence diversity. *Genome Biology* **21**, 164 (2020)
6. Chikhi, R., Limasset, A., Medvedev, P.: Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics* **32**(12), i201–i208 (2016)

7. Cimermancic, P., Medema, M.H., Claesen, J., Kurita, K., Brown, L.C.W., Mavrommatis, K., Pati, A., Godfrey, P.A., Koehrsen, M., Clardy, J., Birren, B.W., Takano, E., Sali, A., Linington R.G., Fischbach, M.A.: Insights into secondary metabolism from a global analysis of prokaryotic biosynthetic gene clusters. *Cell* **158**(2), 412–421 (2014)
8. Cuadrat, R.R.C., Ionescu, D., Dávila, A.M.R., Grossart, H.P.: Recovering genomics clusters of secondary metabolites from lakes using genome-resolved metagenomics. *Front. Microbiol.* **9**, 251 (2018)
9. Donia, M.S., Fischbach, M.A.: Small molecules from the human microbiota. *Science* **349**(6246), 125476 (2015)
10. Eddy, S.R.: Profile hidden Markov models. *Bioinformatics* **14**(9), 755–63 (1998)
11. Eddy, S.R.: Accelerated profile HMM searches. *PLoS Comput. Biol.* **7**, e1002195 (2011)
12. Gao, Y., Li, H.: Quantifying and comparing bacterial growth dynamics in multiple metagenomic samples. *Nature Methods* **15**, 1041–1044 (2018)
13. Hannigan, G.D., Prihoda, D., Palicka, A., Soukup, J., Klempir, O., Rampula, L., Durcak, J., Wurst, M., Kotowski, J., Chang, D., Wang, R., Piizzi, G., Temesi, G., Hazuda, D.J., Woelk, C.H., Bitton, D.A.: A deep learning genome-mining strategy for biosynthetic gene cluster prediction. *Nucleic Acids Res.* **47**(18), e110 (2019)
14. Hyatt, D., Chen, G., LoCascio, P.F. et al.: Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics* **11**, 119 (2010)
15. Kang, D.D., Li, F., Kirton, E., Thomas, A., Egan, R., An, H., Wang, Z.: MetaBAT 2: an adaptive binning algorithm for robust and efficient genome reconstruction from metagenome assemblies. *PeerJ* **7**, e7359 (2019)
16. Korem, T., Zeevi, D., Suez, J., Weinberger, A., Avnit-Sagi, T., Pompan-Lotan, M., Matot, E., Jona, G., Harmelin, A., Cohen, N., Sirota-Madi, A., Thaiss, C.A., Pevsner-Fischer, M., Sorek, R., Xavier, R., Elinav, E., Segal, E.: Growth dynamics of gut microbiota in health and disease inferred from single metagenomic samples. *Science* **349**(6252), 1101–1106 (2015)
17. Lewis, J.D., Chen, E.Z., Baldassano, R.N., Otley, A.R., Griffiths, A.M., Lee, D., Bittinger, K., Bailey, A., Friedman, E.S., Hoffmann, C., Albenberg, L., Sinha, R., Compher, C., Gilroy, E., Nessel, L., Grant, A., Chehoud, C., Li, H., Wu, G.D., Bushman F.D.: Inflammation, antibiotics, and diet as environmental stressors of the gut microbiome in pediatric Crohn’s disease. *Cell Host Microbe* **18**(4), 489–500 (2015)
18. Li, D., Liu, C.M., Luo, R., Sadakane, K., Lam, T.W.: MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* **31**(10), 1674–1676 (2015)
19. Li, H.: Microbiome, metagenomics and high dimensional compositional data analysis. *Annu. Rev. Stat. Appl.* **2**, 73–94 (2015)
20. Lu, J., Breitwieser, F.P., Thielen, P., Salzberg, S.L.: Bracken: estimating species abundance in metagenomics data. *PeerJ Comput. Sci.* **3**, e104 (2017)
21. Ma, R., Cai, T.T., Li, H.: Optimal permutation recovery in permuted monotone matrix model. *J. Am. Stat. Assoc.* Accepted (2020)
22. Marcais, G., Kingsford, C.: A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* **27**(6), 764–770 (2011)
23. McLaren, M.R., Willis, A.D., Callahan, B.J.: Consistent and correctable bias in metagenomic sequencing experiments. *eLife*, article 46923 (2019)
24. Medema, M.H., Blin, K., Cimermancic, P., de Jager, V., Zakrzewski, P., Fischbach, M.A., Weber, T., Takano, E., Breitling, R.: antiSMASH: rapid identification, annotation and analysis of secondary metabolite biosynthesis gene clusters in bacterial and fungal genome sequences. *Nucleic Acids Res.* **39**(2), W339–W346 (2011)
25. Meleshko, D., Mohimani, H., Tracanna, V., et al.: BiosyntheticSPADEs: reconstructing biosynthetic gene clusters from assembly graphs. *Genome Research* **29**(8), 1352–1362 (2019)
26. Menegaux, R., Vert, J.P.: Embedding the de Bruijn graph, and applications to metagenomics. *bioRxiv* 2020.03.06.980979
27. Pasolli, E., Asnicar, F., Manara, S., et al.: Extensive unexplored human microbiome diversity revealed by over 150,000 genomes from metagenomes spanning age, geography, and lifestyle. *Cell* **176**(3), 649–662, e20 (2019)

28. Quince, C., Walker, A.W., Simpson, J.T., Loman N.J., Segata, N.: Shotgun metagenomics, from sampling to analysis. *Nature Biotechnology* **35**(9), 833–844 (2017)
29. Seah, B.K.B., Gruber-Vodicka, H.R.: gbttools: Interactive visualization of metagenome bins in R. *Front. Microbiol.* **6**, 1451 (2015)
30. Sunagawa, S., Mende, D.R., Zeller, G., Izquierdo-Carrasco, F., Berger, S.A., Kultima, J.R., Coelho, L.P., Arumugam, M., Tap, J., Nielsen, H.B., Rasmussen, S., Brunak, S., Pedersen, O., Guarner, F., de Vos, W.M., Wang, J., Li, J., Doré, J., Ehrlich, S.D., Stamatakis, A., Bork, P.: Metagenomic species profiling using universal phylogenetic marker genes. *Nature Methods* **10**, 1196–1199 (2013)
31. Truong, D.T., Franzosa, E.A., Tickle, T.L., Scholz, M., Weingart, G., Pasolli, E., Tett, A., Huttenhower, C., Segata, N.: MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nature Methods* **12**, 902–903 (2015)
32. Wang, S., Cai, T.T., Li, H.: Hypothesis testing for phylogenetic composition: A minimum-cost flow perspective. *Biometrika*. Accepted (2020)
33. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology* **15**, R46 (2014)
34. Ye, S.H., Siddle, K.J., Park, D.J., Sabeti, P.C.: Benchmarking metagenomics tools for taxonomic classification. *Cell* **178**(4), 779–794 (2019)
35. Zhu, Z., Ren, J., Michail, S., Sun, F.: MicroPro: using metagenomic unmapped reads to provide insights into human microbiota and disease associations. *Genome Biology* **20**(1), 154 (2019)

Bioinformatics Pre-Processing of Microbiome Data with An Application to Metagenomic Forensics



Samuel Anyaso-Samuel, Archie Sachdeva, Subharup Guha,
and Somnath Datta

1 Introduction

Samples for environmental microbiome analysis are collected from a variety of surfaces and environments such as plants, soil, ocean, public transit systems, public benches, stairwell handrails, elevators, and urban environments. Analysis that focuses on human microbiome relies on samples from different body sites such as skin, gut, tongue, buccal mucosa, stool, etc. Metagenomic experiments aim to describe microbial communities from these samples using high-throughput DNA sequencing, also known as next-generation sequencing (NGS) technologies. This has further helped scientists around the world to peek into a plethora of diversity of microbes in our environment. The data from these sequencing technologies pose various statistical and computational problems. Also, the sheer magnitude and special data characteristics make metagenomic data analysis a challenging task.

Metagenomic analysis has diverse applications and has led to foundational knowledge on various aspects of human lives. The composition of the human gut microbiome is associated with the physiological and psychological aspects of human health [28, 33, 61, 66, 67]. Metagenomic analysis has a wide-scale application in designing healthy urban environments [47] and discovering novel anti-resistant microbial strains [58]. Metagenomic analysis of microbial communities also provides a significant source of information in forensic science. One of the many questions in forensic studies that metagenomic analysis can answer is predicting the source origin of the metagenomic sample [10, 11, 15]. In this chapter, we discuss various classification methods that can be applied to achieve this goal.

S. Anyaso-Samuel · A. Sachdeva · S. Guha · S. Datta (✉)

Department of Biostatistics, University of Florida, Gainesville, FL, USA

e-mail: sanyasosamuel@ufl.edu; archiesachdeva@ufl.edu; s.guha@ufl.edu;
somnath.datta@ufl.edu

Human Microbiome Project (HMP) [65] and Earth Microbiome Project (EMP) [26] are some of the large-scale initiatives that have offered a comprehensive database for microbiome research. The [MetaSUB](#) Consortium comprised of an international group of scientists is involved in the collection and sequencing of samples from numerous cities in different countries to understand the microbial signature across and within the public spaces of cities around the world. These large-scale data are published by the Critical Assessment of Massive Data Analysis ([CAMDA](#)) in the public domain to find innovative solutions to the pressing questions in modern life sciences. We use the data from CAMDA 2020 Geolocation Challenge and demonstrate a step-by-step approach for metagenomic data analysis. The analysis is divided into two parts, namely, upstream and downstream analysis. In the upstream analysis, we discuss the process of converting raw data of sequenced reads into an $n \times p$ data matrix ready for statistical analysis. This process involves quality control, taxonomic assignment, and estimation of taxonomic abundance of the sequenced reads from different samples. In the downstream analysis, we apply various classification methods and compare their performance for the prediction of the geographical location of microbial samples. Several supervised learning classifiers, such as Support Vector Machines (SVMs), Extreme Gradient Boosting (XGB), Random Forest (RF), and neural networks, can be applied to predict the geolocation of the metagenomic samples. Along with these classifiers, we describe the construction and implementation of an optimal ensemble classification algorithm proposed by Datta et al. [18], which combines several candidate classification algorithms and adaptively produces results that are better or as good as the best classifier included in the ensemble.

2 Bioinformatics Pipeline

2.1 *Microbiome Data*

Microbiome samples are sequenced using next-generation sequencing technologies. The two most widely used sequencing techniques are metataxonomics that use amplicon sequencing of the 16S rRNA marker genes and metagenomics that use random shotgun sequencing of DNA or RNA [8, 45]. Until recently, most studies sequenced the 16S ribosomal RNA gene that is present in bacterial species or focused on characterizing the microbial communities at higher taxonomic levels. Following the drop in the cost of sequencing, metagenomics studies have increasingly used shotgun sequencing that surveys the whole genome of all the organisms including viruses, bacteria, and fungi present in the sample [57].

Metagenomic samples in our case study were sequenced using Illumina HiSeq next-generation shotgun sequencing technology, and the raw data for each sample was obtained in the form of paired-end *.fastq* files with forward and reverse reads. Fastq files contain both nucleotide sequences and their corresponding quality

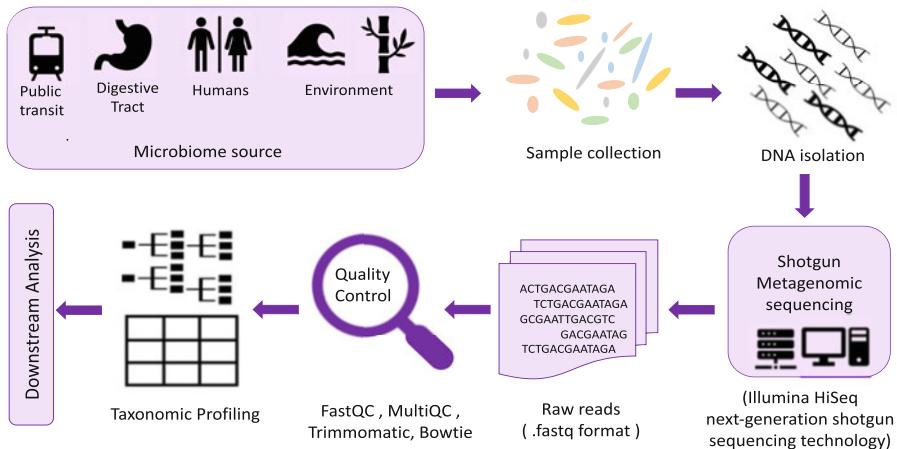


Fig. 1 Schematic representation of the bioinformatics pipeline for metagenomic analysis

scores, known as Phred scores. These scores are used in the quality assessment of these sequencing reads. For the upstream analysis, we started with assessing the quality of the paired-end WGS (whole-genome sequencing) reads followed by their taxonomic classification. Please note the taxonomic classification should not be confused with the city-specific classification that we perform in the downstream analysis. Taxonomic classification refers to mapping the raw sequenced reads of a sample to an existing database of known genomic sequences to produce taxonomic abundance profiles for each sample. Figure 1 shows the schematic representation of the bioinformatics pipeline constructed for the analysis of the metagenomic data. The components of this pipeline are described in detail in the following sections. Table 1 provides information on the data set being analyzed in this chapter. The data set comprised 1065 samples collected from 23 cities around the world.

2.2 Quality Control

Raw NGS reads contain different types of contamination such as low-quality reads, adapter sequences, and host reads. It has been noticed that low-quality sequences can result in misleading inference from the downstream analysis [14, 71]. Hence, it is important to assess the quality of raw sequencing reads before moving ahead with the downstream analysis. If the metagenomic samples are contaminated due to the presence of host (human) sequences, it is necessary to identify and filter out the host reads.

There are a variety of computational tools that can be used for quality control for removing the contaminants and low-quality reads, such as FastQC [2], Cutadapt [46], Trimmomatic [4], and BBTools. The quality of reads from a sample can

Table 1 Frequency of samples from the 23 cities considered in this chapter. The samples were sampled from two collections (CSD16 & CSD17) and obtained by the MetaSUB consortium. The average number of reads was obtained after performing quality control and pre-processing

Location code	Location	Country	# Samples	Avg. # of reads
ARN	Stockholm	Sweden	50	1,621,983
BCN	Barcelona	Spain	38	2,763,249
BER	Berlin	Germany	41	6,095,554
DEN	Denver	USA	45	2,293,732
DOH	Doha	Qatar	65	2,400,540
FAI	Fairbanks	USA	48	6,860,242
HKG	Hong Kong	China	49	3,066,755
ICN	Seoul	South Korea	50	3,053,297
IEV	Kiev	Ukraine	49	2,179,260
ILR	Ilorin	Nigeria	97	10,660,493
KUL	Kuala Lumpur	Malaysia	30	2,310,143
LCY	London	England	37	2,477,320
LIS	Lisbon	Portugal	19	2,864,004
NYC	New York City	USA	99	3,170,947
OFF	Offa	Nigeria	26	22,772,079
SAO	Sao Paulo	Brazil	29	1,989,278
SCL	Santiago	Chile	26	10,399,795
SDJ	Sendai	Japan	32	1,571,323
SFO	San Francisco	USA	29	1,471,680
SGP	Singapore	Singapore	48	2,761,780
TPE	Taipei	China	50	2,755,260
TYO	Tokyo	Japan	75	1,996,146
ZRH	Zurich	Switzerland	33	2,827,183

be assessed by using the diagnostics report generated by FastQC [2], and these quality assessment reports can be further aggregated into a single report using MultiQC [21] for multiple samples. Figure 2 shows the quality score plots from MultiQC for three arbitrarily selected cities from three continents in our study. The *x*-axis shows the positions of the bases, and the *y*-axis represents the Phred score. The Phred score ($= -10 \log_{10} P$) is an integer value representing the estimated probability P of error for identifying the bases generated by DNA sequencing technology. A Phred score of 40 of a base implies that the chance of this base being called incorrectly is 1 in 10,000 [22]. We employed KneadData (version 0.7.4) [49] for quality control analysis. KneadData invokes Trimmomatic [4] for quality trimming, filtering, and removal of adapter sequences. It further calls Bowtie2 [38], which maps the sample reads to a reference human genome database. We discard reads that map to the human genome database. The code snippets below demonstrate how we assessed quality using FASTQC and performed quality control using KneadData. In the pre-QC step, we analyze whether it is necessary to improve the quality of reads. Notice that some of the reads in the second

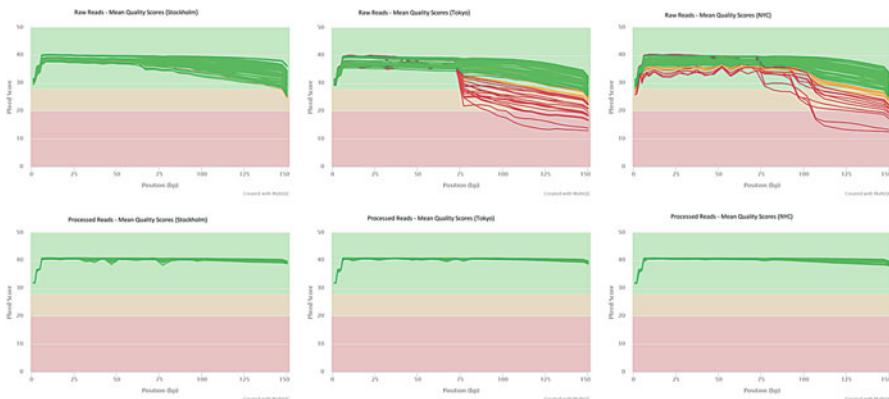


Fig. 2 Aggregated quality score plots from MultiQC for *Stockholm*, *Tokyo*, and *New York City*. The top panel shows the plots for the raw WGS data (pre-QC), while the bottom panel shows the plots for the pre-processed (post-QC) data

and third columns of Fig. 2 have poor-quality scores (below 30). Hence, we choose to trim or drop poor-quality reads. Based on the pre-QC assessment, one can define various rules to improve the quality of the reads to be used for subsequent analysis. For example, in the quality control code, the parameter `ILLUMINACLIP:NexTERAPE-PE.fa:2:30:10:8:keepBothReads` `SLIDINGWINDOW:4:30 MINLEN:60` prompts Trimmomatic to remove adapters, defines a sliding window that cuts a read once the mean quality in a window of size 4 falls below a Phred score of 30, and retains sequences with a minimum length of 60. This procedure results in sequencing reads with reasonably good quality. We assessed the quality of the reads after quality control using MultiQC and noticed an obvious improvement in the quality of the reads when compared to the raw reads. The upper panel of Fig. 2 shows the reports from pre-QC analysis, and the lower panel of Fig. 2 shows the plots from the post-QC analysis. The code below can be used as a basic guideline for performing the bioinformatic pre-processing of raw sequenced reads. We encourage readers to make appropriate modifications to the parameters of the bioinformatics tools to suit the goal of their analysis. These tools are also constantly undergoing development. Consequently, it is recommended that the researcher works with the most recent versions of software and databases used for sequence mapping.

Pre-QC analysis

```
# make a folder to store FastQC output
$ mkdir output_folder
# Perform quality control checks on the samples using FastQC
$ module load fastqc/0.11.7
$ fastqc -t 30 *.fastq.gz -o output_folder/
# Aggregate the results of fastqc quality control checks using MultiQC
```

```
$ cd ~/output_folder
$ module load multiqc/1.7
$ multiqc *_fastqc.zip
```

Quality Control

```
$ module load kneadata/0.7.4
$ module load bowtie2/2.3.5.1
$ mkdir KneadData_output_folder
#####
# Download Trimmomatic and adapter sequence files
$ curl -LO http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/
Trimmomatic-0.36.zip
$ unzip Trimmomatic-0.36.zip
#####
# Download the Homo_Sapiens database
$ mkdir DB_folder
$ cd ~/DB_folder
$ kneadata_database --download human_genome bowtie2 ./
#####
# Use a loop for analysis of multiple gzipped paired-end reads using kneadata
$ for f in $(ls *.fastq.gz | sed -e 's/_1.fastq.gz//' -e 's/_2.fastq.gz//' | \
sort -u)
$ do
$     echo "Unzips ${f}"
$     gzip -d -f ${f}_1.fastq.gz > ${f}_1.fastq
$     gzip -d -f ${f}_2.fastq.gz > ${f}_2.fastq
$     echo "Preprocessing ${f}"
$     kneadata -i ${f}_1.fastq -i ${f}_2.fastq -o KneadData_output_folder \
-db /path/to/DB_folder --trimmomatic /path/to/Trimmomatic-0.36 -t 30 \
--trimmomatic-options "ILLUMINACLIP:/path/to/adapter/sequence/file
:2:30:10:8:keepBothReads SLIDINGWINDOW:4:30 MINLEN:60" \
--bowtie2-options "--very-sensitive --dovetail"
$     echo "Completed QC for ${f}"
$ done
```

Post-QC analysis

```
$ cd /path/to/KneadData_output_folder
$ mkdir fastqc_output_folder
$ module load fastqc/0.11.7
$ fastqc -t 30 *paired* -o fastqc_output_folder/
$ module load multiqc/1.7

# Aggregate the results of fastqc quality control checks
# using MultiQC
$ cd /path/to/fastqc_output_folder
$ multiqc *_fastqc.zip
```

2.3 Taxonomic Profiling

After quality control of the sequencing reads, the next step is to estimate the taxonomic abundance of each sample. A taxonomic abundance table is an $n \times p$ matrix of absolute or relative abundance of p identified taxa in n samples. Taxonomic profiling of sequenced reads typically comprises two steps. First, the classification or the alignment of sequence reads to a database of microbial genomes. The second step involves the estimation of the abundance of each taxon (species, genus, etc.) in the metagenomic sample, i.e., estimating the number or percentage of reads belonging to each taxon. Various algorithms and tools have been developed to efficiently classify sequencing reads to known taxa with improved speed [70]. A variety of metagenomic profiling tools match sequences to known databases. These databases created at different times may have different contents as they go through regular updates with the addition of new sequences.

Taxonomic profiling tools use a variety of approaches such as alignment of marker genes (MetaPhlAn2 [59], mOTU [63], GOTTCHA [23]), k -mer mapping in WGS reads (Kraken [23], CLARK [53]), translating DNA into amino acid sequences, and mapping to protein databases (Kaiju [50], DIAMOND [9]).

This chapter does not pursue the goal of reviewing all of these taxonomic profiling tools. Several research papers provide discussion on the review and the comparison of these taxonomic profiling tools [3, 8, 43, 48]. Performance is usually compared on the basis of the proportion of mapped reads, run time, sensitivity, and other performance metrics. Since the evaluation of these tools is a complex task, no single metric is usually used to judge the performance; rather, multiple factors are examined. Considering that some tools utilize a limited set of marker genes while others use expansive databases, judging a profiling tool only by the proportion of reads mapped may not be adequate [43]. Since the application of any taxonomic profiling tool will potentially impact the results and conclusions of the metagenomic study, the selection of the appropriate tool should be based on performance metrics that suites the analyst's scientific investigation. In this section, we describe and also discuss the implementation of three commonly used taxonomic profiling tools, namely MetaPhlAn2, Kraken2, and Kaiju.

2.3.1 MetaPhlAn2

We implement MetaPhlAn2 [59] for the quantitative taxonomic profiling of our quality-controlled sequenced reads. MetaPhlAn2 is computationally fast as it relies on the clade-specific marker genes approach for taxonomic profiling [59], and this approach is not expected to map all reads. Taxonomic assignment is attained by aligning the sequence reads to the marker set using Bowtie2 [38]. In the application, we used the default settings of MetaPhlAn2 to extract species-level relative abundances for each sample, and these values lie within [0, 1]. The relative abundances for each sample were then merged into a large relative abundance table

using a custom MetaPhlAn2 script. After termination of MetaPhlAn2 procedure, we obtained a table of relative abundances of 1049 species for 1047 samples. In this setup, we have chosen to obtain species-level relative abundance. However, information for other taxonomic levels can be easily extracted from the output generated by MetaPhlAn2.

Taxonomic Profiling with MetaPhlAn2

```
# Perform taxonomic profiling of multiple .fastq files using MetaPhlAn2
$ module load metaphlan2/2.96.1
$ for f in $(ls *.fastq | sed -e 's/_1.fastq//' -e 's/_2.fastq//' | \
  sort -u)
$   do
$     metaphlan2.py --bowtie2db /path/to/metaphlan_databases \
      ${f}_1.fastq,${f}_2.fastq --bowtie2out ${f}.bt2out \
      --nproc 30 --input_type fastq
$   metaphlan2.py --bowtie2db /path/to/metaphlan_database ${f}.bt2out \
      --nproc 30 --input_type bowtie2out > ${f}_profile.txt
$ done

# Merge taxonomic profiles for each sample into a single .txt file
$ merge_metaphlan_tables.py *_profile.txt > merged_abundance_table.txt
```

2.3.2 Kraken2

Kraken2 [69] is a rapid and highly accurate metagenomic classification tool that uses a k -mer approach. For assignment of sequence reads to taxonomic labels, it utilizes the k -mer information within each read, and each k -mer is mapped to the lowest common ancestor (LCA) of the genomes that contains the k -mer in a custom-built database. Lu et al. [44] point out that the LCA approach employed by the Kraken system means that the system is likely to underestimate the number of reads that are directly classified as species.

To overcome the issue of underestimation of taxonomic abundance by the Kraken system, Bracken [44] was developed. Bracken uses a Bayesian algorithm and the results from the Kraken2 classification for estimation of the relative abundance of a metagenomic sample at the user-specific taxonomic level. To illustrate the difference between these tools, the developers of Bracken report an instance [44] that we consider here. The genomes of *Mycobacterium bovis* and *Mycobacterium tuberculosis* are 99.95% identical. Since these species are very similar, Kraken classifies the vast majority of reads from either of them to their LCA, which in this case is the genus *Mycobacterium*. On the other hand, Bracken uses information on some reads from the species-specific portion of the genome along with the similarity information between close species to move reads from the genus level to the species level.

To estimate the abundance for each sample using the Kraken2–Bracken system, we employed a pre-computed standard database that consists of reference sequences from archaea, bacteria, and the human genome. Further, to generate the Bracken database file, the switch `-t` indicates the number of threads to use, and `-l` indicates the read length of the data. Since most of our data were 150 base pair (bp) reads, we set `-l` to be 150, and we use the default k -mer length of 35. Then, for each paired-end sample, we generate reports from the Kraken2 taxonomic assignment procedure, and these reports are then passed into the Bracken program for abundance estimation. Estimation of the abundance was carried out at the species level (`-l S`), with a default reads threshold of 10 (`-t 10`). Finally, we use a custom script to combine the Bracken output for all samples into a large single file. The column of interest in the Bracken output is the `new_est_reads`, which gives the newly estimated reads. After obtaining the abundance table, normalization was carried out using the cumulative sum scaling approach. This procedure was implemented with the `metagenomeSeq` [55] R package.

Taxonomic Profiling with Kraken2–Bracken

```
# loads kraken2 & bracken
$ module load kraken/2.0.8b bracken/2.5
# Generates the bracken database file
$ bracken-build -d /path/to/kraken2/database -t 30 -k 35 -l 150 \
    -x /path/to/kraken2/installation/directory
$ echo "Building bracken database file complete"
$ cd /path/to/pair-end/.fastq/files
# Run Kraken2 & Bracken for abundance estimation
$ for f in $(ls *.fastq.gz | sed -e 's/_1.fastq.gz//' -e | \
    's/_2.fastq.gz//' sort -u)
$ do
# Generate kraken2 report files
$ kraken2 --db /path/to/kraken2/database --threads 30 --report
${f}.kreport \
    --fastq-input --gzip-compressed --paired ${f}_1.fastq.gz ${f}_2.
    fastq.gz \ > ${f}.kraken
# Estimate abundance with Bracken
$ bracken -d /path/to/kraken2/database -i ${f}.kreport -o ${f}.bracken
    -r 150 \ -l S -t 10
$ done
$ echo "Estimation of species abundance with kraken2-bracken complete"
# Combining bracken output files
$ cd /path/to/.bracken/files
$ combine_bracken_outputs.py --files *.bracken -o output_file
```

2.3.3 Kaiju

For the given DNA sequences, Kaiju [50] translates the reads into amino acid sequences and compares these reads against a reference database of protein sequences. It creates an efficient database structure by indexing the reference protein database using the Burrows–Wheeler transform (BWT) and saves each

sequence in an FM-index (Full-text index in Minute space) table. It then searches for maximum exact matches between the reads and the reference database created. Kaiju's developers [50] emphasize that protein-level classifiers such as Kaiju are more sensitive to novel or highly variable sequences because protein sequences are more conserved than the underlying DNA. Moreover, protein sequences are more tolerant to sequencing errors due to the lower mutation rate of amino acid sequences as compared with nucleotide sequences [1, 70].

To execute the taxonomic classification of sequencing reads using Kaiju, we used nr database as our reference database. Program `kaiju-makedb` downloads the source database of interest and constructs Kaiju's index using BWT and FM-index. We observed that some tools used for quality control of the sequences may create disorder in the read names in both `.fastq` files. If the read names are not identical between the first and second files, program `kaiju` issues an error. We used `Repair` function from `bbmap` to fix this issue before moving ahead with the taxonomic classification of sequencing reads. For faster implementation, we used `kaiju` with multiple parallel threads using option `-z 25` in MEM mode (`-a mem`). The output files obtained from program `kaiju` comprised 3 columns, classification status C/U for each read, read names, and NCBI taxon identifier of the assigned taxon. These output files were further summarized into a table using `kaiju2table` script, which gives read count (or percentage) for all samples and taxa in a long format. To process this data for the downstream analysis, we converted it into a wide format with taxa as rows and samples as columns using `pivot_wider` function from `tidyverse` package in R. Users may also choose to run Kaiju in greedy mode that yields a higher sensitivity as compared to the MEM mode, sometimes at the cost of increased run time.

Taxonomic Profiling with Kaiju

```
# load kaiju
$ module load kaiju/1.7.2
$ module load bbmap
# Create reference database index
$ kaiju-makedb -s nr
# Repair disordered paired-end files
$ mkdir bbmap_ordered
$ cd /path/to/fastq/files
$ for f in $(ls *.fastq.gz | sed -e 's/_1.fastq.gz//' -e 's/_2.fastq.gz//' | sort -u) do
$   repair.sh in1=${f}_1.fastq.gz in2=${f}_2.fastq.gz \
$     out1= bbmap_ordered/${f}ORDERED_1.fastq.gz out2= bbmap_ordered/${f}ORDERED_2.fastq.gz \
$     outs=bbmap_ordered/${f}ORDERED_singleton.fastq.gz repair
$ done
$ cd bbmap_ordered
$ rm *ORDERED_singleton.fastq.gz
# Run Kaiju to assign reads to taxa
$ mkdir TaxoClassn
# start - taxonomic classification
for f in $(ls *.fastq.gz | sed -e 's/_1.fastq.gz//' -e 's/_2.fastq.gz//' | sort -u)
  do
$   kaiju -z 25 -t /path/to/kaijus/database/Directory/nodes.dmp \
$     -f /path/to/kaijus/database/Directory/kaiju_db_nr.fmi \
$     -i ${f}_1.fastq.gz -j ${f}_2.fastq.gz -o ${f}.out -a mem
$ mv ${f}.out TaxoClassn
$ done
```

```
# Create summary table of the output files at a taxonomic rank
# Merge files from all samples to a single table
$ cd TaxoClassn
$ kaiju2table -t /path/to/kaijus/database/Directory/nodes.dmp \
-n /path/to/kaijus/database/Directory/names.dmp \
-r species -o Merged_files.tsv *.out \
-c 10 -l superkingdom,phylum,class,order,family,genus,species
```

As mentioned earlier, the choice of profiling tool may depend on multiple factors such as classification speed, proportion of mapped reads, output format, ease of use, and computational resources available. If the analyst has access to good computational resources with high amounts of available memory (>100Gb), then Kraken, Bracken, and Kaiju are useful options. If sufficient computational resources are not available, then MetaPhlAn is a viable alternative with fast classification speed. Kaiju, for instance, has a [web server](#) where one can upload the compressed *fastq* files and select different options for taxonomic assignment for an easier implementation without running bash scripts via the command line. Simon et al. [70] provide an interesting and informative assessment of the performance of several metagenomic tools used for taxonomic profiling of real and simulated data sets.

2.4 Computing facilities

All bioinformatics procedures were performed using the University of Florida HiPerGator2 supercomputer. HiPerGator2 has 30,000 cores in Intel E5-2698v3 processors with 4 GB of RAM per core, and a total storage size of 2 petabytes (PB). Bash scripts and *.fastq* files were stored on the supercomputer's parallel file system that offers high performance for data analysis. For the computing jobs submitted to the cluster, we typically requested an allocation of a single computing node, 20 cores per task, and 200 GB memory.

3 Methodology

In Sect. 2.3, we discussed several techniques for taxonomic profiling that comprised taxonomic classification/assignment and estimation of abundance. At the termination of each profiling technique presented, we obtained a species abundance table. Now, the rest of this chapter will focus on methods for classifying taxa abundances to known class labels. That is, we pursue the goal of modeling taxa abundances of metagenomic samples belonging to known class labels. Then, the model is used to predict class labels for new metagenomic samples based on their estimated abundances. For our analysis, the class labels are the source cities where samples originated. The classification of sequence reads to taxonomic labels should not be

mistaken for the classification of abundance profiles to source cities. We stress that the term classification will refer to the latter described herein.

As we indicated in the previous paragraph, this section focuses on the supervised learning analysis of the pre-processed metagenomics data. We highlight methods for feature selection, present several classification algorithms that include the ensemble classifier, discuss techniques to overcome the problem of class imbalance, and finally discuss measures for evaluation of model performance.

3.1 Pre-Processing and Feature Selection

The species abundance matrix obtained after the taxonomic profiling contains a large set of features, i.e., taxa. For instance, 6152 taxa were obtained after taxonomic profiling with the Kraken2–Bracken system, while 1049 taxa and 32,146 taxa were obtained after profiling was, respectively, performed with MetaPhlAn2 and Kaiju.

Similar to the cases presented here, the most abundance data obtained from metagenomics samples are high-dimensional in nature, and it is usually desirable to extract only important features from the data. Common feature reduction techniques are based on the prevalence of the taxa in the abundance table. For instance, taxa with less than a specified number of reads, say 10, can be dropped. In addition, taxa that are present in less than, say, 1% of the samples may also be discarded. If these approaches are employed, then the resulting abundance table should be re-normalized.

Other advanced methods exist for feature selection, and in this section, we describe a couple of these techniques. In practice, feature selection aims at obtaining a reduced subset of relevant informative features that bolster the assignment of samples of known class labels based on their abundance information. However, from our experience and those of several research studies [54], feature selection may not provide a substantial improvement in the predictive ability of the fitted classification models due to the complex nature of microbiome data. Hence, even though fitting classification models on the data with a reduced feature space may be more computationally efficient, we recommend that analysts should also investigate the performance of such models when trained on the data with a complete feature space.

Among the other approaches to feature selection, first, features could be selected based on the importance scores returned after a supervised training of the Random Forest model on the data with a complete set of features. The features are ranked according to their importance scores, and the top k features are chosen as the set of informative features. The classification model of interest is then trained with the k selected importance features. In this setup, k is usually chosen from a set of a predetermined number of features via cross-validation, such that the number of features from the predetermined set that maximizes classification accuracy is chosen to be k . Pasolli et al. [54] utilized this method in their review study that assessed machine learning approaches for metagenomics-based prediction tasks.

In another heuristic approach, one may choose to use the Lasso [64] or ElasticNet [73] with a multinomial model for feature selection. However, the standard versions of penalized regression methods are not efficient for the analysis of relative abundance data because of the compositional nature of the data [26]. Owing to this fact, regression [31] and variable selection methods [41], which impose sum-to-zero constraints for the Lasso, have been developed for compositional data.

The hierarchical feature engineering (HFE) [52] technique is a recently developed tool for performing feature selection. To obtain a smaller set of informative microbial taxa, this tool uses information from the taxonomy table, the correlation between taxonomic labels, and the abundance data to exploit the underlying hierarchical structure of the feature space. At the termination of the algorithm after analyzing a species abundance table, it returns an OTU table that contains a combination of both species and other higher-level taxa. Fuzzy [19] is another modern tool for feature selection. It is a collection of functions for performing widely implemented feature selection methods such as the Lasso, information-theoretic methods, and the Neyman–Pearson feature selection approach. Developers of the HFE used the predictive performance of several machine learning models to compare the HFE with other standard feature selection tools that do not account for the hierarchical structure of microbiome data. They reported that the HFE outperformed the other methods.

3.2 *Exploration of Candidate Classifiers*

In this section, we present brief descriptions of some supervised learning models commonly used for the classification of abundance values of metagenomics samples to known class labels. Our survey of algorithms will largely focus on supervised classifiers that are suitable for analyzing multiclass classification problems. These classifiers can be broadly partitioned into linear and non-linear classifiers.

Linear methods for classification such as linear discriminant analysis, quadratic discriminant analysis, regularized discriminant analysis, logistic regression, and SVM (without kernels) achieve classification of objects based on the value of a linear combination of features in the training data. These classifiers solve classification problems by partitioning the feature space into a set of regions that are defined by class membership of the objects in the training data. Also, the decision boundaries of the partitioned regions are linear [32]. Generally, these classifiers also take less time to train than non-linear classifiers. However, by using the so-called kernel trick, some linear classifiers can be converted into non-linear classifiers that operate on a different input scale.

In cases where the training data are not linearly separable (usually via a hyperplane), a linear classifier cannot perfectly distinguish classes of such data. For such cases, the non-linear classifiers will often provide better classification performance than the linear classifiers. Examples of non-linear classifiers commonly

used for classification in metagenomics studies include the *kernel* SVM, Random Forest (RF), and neural networks (multilayer perceptron):

- **Recursive Partitioning (RPart)**—A decision tree [7] is the fundamental element of the RPart model. A decision tree is grown by repeatedly splitting the training data set into subsets based on several dichotomous features. The recursive splitting from the root node to the terminal node is based on a set of rules determined by the features. The process is recursive in nature because each subset can be split an indefinite number of times until the splitting process terminates after a stopping criterion is reached. In the case where the target response is a unique set of labels, the tree model is called a classification tree. For the prediction of the class label of a new subject, the model runs the observation from the root node to the terminal node that assigns the class membership.
- **Random Forests (RF)**—The idea of the RF classifier [6] is to grow a collection of trees by randomizing over subjects and features. That is, each tree in the forest is grown by using a bootstrap sample from the training data. Out-of-bag samples comprise samples that are not included in the bootstrap sample. These samples serve as a validation set. In contrast to *bagging* that uses all p predictors for splitting at each node, RF uses only $m < p$ randomly selected features to obtain the best split. With the implementation of this step, the correlation between the trees is reduced. Also, it improves the classification performance obtained when a *bagging* procedure is implemented. Unlike decision trees, no pruning is performed for Random Forests, i.e., each tree is fully grown. For predicting the class of a new observation, each tree in the forest gives a class assignment, and majority voting is used to obtain the final prediction. Advantages of the RF include its robustness to correlated features, its applicability to high-dimensional data and the ability to handle missing data internally in an effective manner, and its use as a feature selection tool through its variable importance plot. Also, it offers competitive classification accuracy for most problems with little parameter tuning and user input.
- **Adaptive Boosting (AdaBoost)**—In the boosting [24] procedure, many weak classifiers are sequentially combined to produce a strong learner. The procedure achieves this by repeatedly training many weak learners on modified versions of the data set, and then the strong learner is created by a weighted average of the weak classifiers. Note that a weak classifier is a learner whose performance is only slightly better than random guessing. Also, the weights used to fit each of the weak classifiers are functions of the prediction accuracy using some previous versions of the weak classifier. If we let $G_m(x)$, $m = 1, \dots, M$ denote a sequence of weak classifiers trained with weighted versions of the training data, the final output of the AdaBoost classifier is a weighted sum of $G_m(x)$. In this case, weights w_i , $i = 1, \dots, N$, that are updated iteratively are applied to the observations in the training set. At the first boosting iteration, $m = 1$, a base classifier, i.e., $w_i = \frac{1}{n}$, is trained. Then, for $m = 2, \dots, M$, observations that were misclassified in the preceding iteration are given more influence than observations that were correctly classified. In this sense, the boosting procedure

is adaptive because each subsequent classifier in the sequence is thereby forced to tweak its performance to favor observations that were misclassified by previous classifiers.

- **Extreme Gradient Boosting (XGBoost)**—Gradient boosting [25], also referred to as gradient boosting machines (GBM), is another boosting algorithm that creates a strong learner from an ensemble of weak classifiers, typically decision trees. In the implementation, this machine combines a gradient descent optimization procedure with the boosting algorithm. The machine is constructed by fitting an additive model in a forward stage-wise manner. A weak learner is sequentially introduced at each stage to improve the performance of existing learners in classifying previously misclassified observations. These misclassified observations are determined by gradients, which in turn guide the improvement of the model. For XGBoost [13], trees are grown to have a varying number of terminal nodes. Contrasting to GBM that employs gradient descent, XGBoost employs Newton boosting that uses Newton–Raphson’s method to obtain the solution to the optimization problem. With set parameters, the XGB algorithm reduces the correlation among the trees grown, thus increasing classification performance. Further, the algorithm utilizes parallel and distributed computing that speeds up learning and enables quicker model exploration. Historically, this classifier has been popular among winning teams participating in machine learning competitions [13].
- **Support Vector Machines (SVM)**—To understand the concept of the SVM [17], first, we consider a binary classification problem for which we intend to assign a new data point to either of two classes. The data point is treated as a p -dimensional vector, and the SVM algorithm aims at finding a $(p - 1)$ -dimensional hyperplane that represents the largest separation between the two classes. Several hyperplanes may exist for partitioning the data. SVM selects the decision boundary that maximizes the distance to the nearest data point on each of its sides as the optimal hyperplane. SVMs are popular for solving classification problems because in the case where no linear decision boundary exists, they can allow for non-linear decision boundaries using the so-called “kernel trick.” Also, SVM solves a multiclass classification problem by decomposing the problem into multiple binary classification problems. In this sense, most SVM software constructs binary classifiers that distinguish between one of the class labels and the others (one-versus-all) or between every pair of classes (one-versus-one). In the latter approach, $\frac{k(k-1)}{2}$ binary classifiers are constructed if the target variable is comprised of k classes. For the prediction of a new observation in the one-versus-all case, the binary classifier with the maximum output function decides the class label, while a majority voting strategy is used to assign the class label in the one-versus-one case.
- **Multilayer Perceptron (MLP)**—Under the deep learning framework, MLP [32] is an interconnected network of neurons or nodes that have weights attached to the edges of the network. MLP utilizes an adaptive mathematical model that changes based on the information that is fed into the network. Using several layers of units, the network maps the input data to an output vector with length

equal to the number of classes in the target variable. First, the input data are passed into an input layer. This layer emits a weighted output that is further passed into another hidden layer of units (there can be more than one hidden layer). In the final branch of the process, the output layer receives the weighted output from the hidden layer and assigns the network's prediction.

Several classifiers provide the option to scale the features so that they have the same variance. This scaling procedure will destroy the compositional nature of the data, and hence, we suggest that scaling should not be done. From the documentation of most classification learning software, we can set the logical `scale` or `standardize` parameters that indicate whether scaling should be carried out. This parameter should be set to `FALSE`.

3.3 *The Ensemble Classifier*

In Sect. 3.2, we presented a variety of popular machine learning models that can be used to predict the source origin of metagenomics samples. These classifiers have been used to analyze data obtained from several experimental studies that aimed to explore associations between microbial imbalance and disease or environmental factors. The RF and SVM classifiers remain state-of-the-art for metagenomics-based classification purposes. In contrast, classifiers such as the AdaBoost and XGBoost that are based on boosting algorithms have not gained much traction in the metagenomics data classification.

Research papers such as Knights et al. [35], Moitinho-Silva et al. [51], and Zhou et al. [72] provide a review of a variety of supervised machine learning models commonly used for feature selection and classification in microbiota studies. The reviews on the classification of microbiota data often report microbiome–phenotype associations and host-microbiome and disease associations. Among other findings, several individual studies have utilized different pre-processing and analysis methods that yielded discrepant conclusions and difficulty of classification models to be generalized across research studies [20, 54, 72].

In the context of exploring the relationship between microbial samples and environmental factors, CAMDA had organized the Metagenomics Geolocation Challenge over the last three years. Participants who have worked on these challenges have used a combination of bioinformatics and machine learning techniques to build microbiome fingerprints for the prediction of the source origins of microbial samples. Neural networks, RF, and SVM are among commonly used machine learning techniques for the construction of such fingerprints. In particular, no single classifier has shown to give consistent optimal performance across these metagenomics studies. When addressing results from a classification competition based on proteomics data, Hand [29] points out this observation as well.

Several reasons may account for the inconsistencies and non-generalizability of machine learning models across microbiome studies. Potential factors that can

elicit inconsistencies in microbiome studies include the nature of the data being studied, sample collection strategies, different sequencing techniques, and varying bioinformatics procedures. Furthermore, the performance of machine learning models is likely to depend on the techniques utilized during the pre-processing and taxonomic profiling of the microbial samples. In practice, it is generally impossible to know *a priori* which machine learning model will perform best for a given classification problem and data.

To create a more robust classifier, Datta et al. [18] proposed an ensemble classifier that combines a variety of classification algorithms in conjunction with dimension reduction techniques (if necessary) for classification-based problems. The ensemble classifier is constructed by bagging and weighted rank aggregation [56], and it flexibly combines the standard classifiers to yield classification performance that is at least as good as the best performing classifier in the set of candidate classifiers that define the ensemble. For any data set under investigation, the ensemble classifier excels in the sense that it adaptively adjusts its performance and attains that of the best performing individual performance without prior knowledge of such classifier(s). Hand [29] also states that the aggregation of results obtained from many fitted models serves to smooth the ensemble model away from a single model that is optimized on the training set, and therefore, the combination of models serves a role similar to regularization.

The ensemble classifier is itself a classification algorithm, and here, we describe the construction of this classifier. Consider the abundance matrix $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_p)$ for n samples and p taxa, where each \mathbf{x}_j , $j = 1, \dots, p$, is normalized, and the target labels, $\mathbf{y} = (y_1, \dots, y_n)$. The steps to build the ensemble classifier are as follows:

1. Choose M candidate classifiers and K performance metrics. Then, for $b = 1, \dots, B$:
 - (i) Draw a bootstrap sample $\mathbf{Z}_b^* = (\mathbf{X}_b^*, \mathbf{y}_b^*)$ of size n for the training data. Ensure samples from all classes are represented in \mathbf{Z}_b^* . OOB samples comprise all samples not included in \mathbf{Z}_b^* .
 - (ii) Train each M classifier with the bootstrapped sample, \mathbf{Z}_b^* .
 - (iii) Use each M classifier to predict the OOB samples.
 - (iv) Based on the true values of the OOB set, and the predicted class labels, compute the K performance measures.
 - (v) Perform weighted rank aggregation: The performance measures used in step (iv) rank the classifiers according to their performance under each measure, thereby producing K ordered lists, L_1, L_2, \dots, L_K , each of size M . Using weighted rank aggregation, the ordered lists are aggregated to determine the best single performing classifier denoted as $A_{(1)}^b$.

The ensemble is a set of $\{A_{(1)}^1, \dots, A_{(1)}^b, \dots, A_{(1)}^B\}$ classifiers.

Notice that the algorithm evaluates the performance of each candidate classifier based on their prediction of the OOB samples. This protects the ensemble classifier from overfitting. Just like cross-validation, the classification performance based on

the OOB samples is estimated using data that were not used when training the classifier. The OOB errors are not the same as the cross-validation errors, but in practical terms, they should be approximately close.

Given the abundance of a new sample, \mathbf{x}_{1xp} , the ensemble classifier gives prediction for such sample using the following procedures:

1. Each classifier, $A_{(1)}^1, \dots, A_{(1)}^B$, in the ensemble is used to predict the class label of \mathbf{x}_{1xp} . Let $\hat{y}_1, \dots, \hat{y}_B$ denote the class predictions from the B models in the ensemble.
2. The final prediction is obtained by majority voting, that is, the most frequent class label among the B predicted classes.

3.4 Class Imbalance

More often than not, metagenomics data are imbalanced. That is, at least one of the classes in the data is underrepresented. Data imbalance is likely to skew the performance of the classification models such that the models will be biased toward the majority classes. For instance, if there are disproportionately more samples from class A than there is from class B , the classification model is prone to assign a random label to class A than class B . Since classification algorithms aim to reduce the overall misclassification rate, rather than the error rate in majority classes, such models will not perform well for imbalanced data. Generally, classification algorithms are poised to perform better with *nearly* equal representation of classes in the training set.

The problem of class imbalance has received considerable attention in the machine learning literature, and a variety of methods exist to mitigate this problem. Some of these methods have also found application in the analysis of metagenomics data. In this section, we briefly describe the underpinnings of such procedures along with their pros and cons. The application of these methods does not improve the overall fit of the classification model discussed. When implemented, they aim to improve the prediction of samples in the minority classes. Roughly speaking, these methods are partitioned into down-sampling, over-sampling, hybrid, and weighting techniques:

- (i) Down-sampling techniques: This involves randomly removing samples from the majority classes until class frequencies are roughly balanced. One disadvantage of this technique is the loss of information in the majority classes since a large part of the majority classes will not be used to train the classifier.
- (ii) Over-sampling techniques: This involves the random replication of samples in the minority classes to attain approximately the same sample sizes in the majority classes. As noted by Chen et al. [12], more information is not added to the data by over-sampling; however by replication, the weight of the minority classes is increased. From our experience, down-sampling appears to be more computationally efficient since the classifier is trained on smaller data sets.

- (iii) Hybrid techniques: This class of techniques combines both over-sampling and down-sampling to artificially create a balance in the data set. For instance, SMOTE (and its variants), AdaSyn, DSRBF, and ProWSyn methods generate synthetic samples from the minority classes to balance class frequencies. Kovács [36] studied the performance of a variety of minority over-sampling techniques when applied to large imbalanced data sets. They report that no over-sampling technique gives consistent optimal performance. Hence, they suggest careful investigation when choosing the technique to use. The `smote-variants` [37] package provides Python implementation for a host of these hybrid techniques, while the `UBL` [5] package provides certain implementations in R. In the context of the analysis of microbiome data, a variety of user-specific hybrid over-sampling techniques have been employed. For instance, Knights et al. [35] used an artificial data augmentation approach to boost the representation of samples when analyzing microarray data. In their approach, they generate noisy replicates by adding a small amount of Gaussian noise to the OTU counts in each sample, with a threshold of zero to avoid negative values. The authors found that the difference in predicted error between their augmented and unaugmented model was at most 2% decrease in error. Also, Harris et al. [31] report an increment in classification accuracy from 83% to 91% after application of an optimized sub-sampling technique to address the problem of data imbalance in their analysis of metagenomics data aimed at predicting sample origins.
- (iv) Weighting: A cost-sensitive approach to fitting classification models is to train them using class weights. In this approach, the algorithms place heavier weights on the minority classes and will penalize the classifier for misclassifying the minority classes. The weighted Random Forest [12] is an example of a classification model that implements class weighting.

To avoid overfitting the data, these techniques for addressing class imbalance are generally applied only to the training set. Further, if a resampling technique (bootstrap or cross-validation) is used for model evaluation during analysis, the over-sampling procedure should be performed inside the resampling technique. This approach is followed because if an over-sampling is done before, for instance, cross-validation is performed, the model is likely to have glanced at some samples in the hold-out set during model fitting; therefore, the hold-out set is not truly unknown to the model. This implementation will result in overly optimistic estimates of model performance.

3.5 Performance Measures

In this section, we focus on measures used for evaluating the performance of classification algorithms on imbalanced data. In such scenarios, the overall classification accuracy is often not an appropriate measure of performance since rare

classes have little impact on accuracy than majority classes [34]. Other performance metrics such as recall (or sensitivity), precision (or positive predictive value, PPV), F-measure, and G-mean are commonly used single-class metrics in binary classification problems. These metrics can also be used to assess the prediction of individual class labels in multiclass problems. These metrics are defined as the following:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

For evaluating the overall performance of the classifiers for imbalanced learning, multiclass extensions of the G-mean [62] and AUC [30], as well as Cohen's Kappa [16], are commonly used metrics.

$$\text{G-mean} = \left(\prod_{i=1}^K \text{Recall}_i \right)^{\frac{1}{K}},$$

$$\text{MAUC} = \frac{1}{K(K-1)} \sum_{i=1}^K \sum_{j \neq i}^K \text{AUC}(i, j),$$

$$\kappa = \frac{P_0 - P_E}{1 - P_E},$$

where K is the number of classes, Recall_i is the recall for class i , P_0 is the relative observed agreement among classifiers (i.e., the overall accuracy of the model), and P_E is the probability that agreement is due to chance. G-mean is the geometric mean of recall values for all classes, while MAUC is the average AUC for all pairs of classes. Apparently, the G-mean will be equal to 0 if the recall for any class is 0. These three performance measures were used in constructing the ensemble classifier that will be implemented in our analysis.

3.6 Data Analysis

In this section, we lay out some analytical techniques for the pre-processed species abundance table. These techniques focus on training supervised machine learning models for the classification of the OTU abundance to known class labels. Here, our analysis will be based on the species abundance tables obtained after bioinformatics

pre-processing and taxonomic profiling of the WGS data gotten from the 2020 CAMDA Forensic Challenge, see Sect. 2.1. In Sect. 2.3, we used three different taxonomic profiling tools to obtain the species abundance table, and the supervised algorithms for classification will be applied to each data set. The primary objective of the analysis lies in predicting the source origins of given metagenomics samples from 23 cities across the globe.

First, we fitted ten candidate classifiers. The candidate classifiers consist of all classifiers discussed in Sect. 3.2 together with certain modifications of these classifiers. For instance, we considered the RF classifier with principal component terms (denoted as PCA+RF) and partial least squares terms (PLS+RF). And we also trained the AdaBoost, XGBoost, RPart classifiers each with PLS terms (PLS+ADA, PLS+XGB, PLS+RPart). Furthermore, we trained the ensemble classifier for which the ensemble constitutes the mentioned candidate classifiers. Candidate classifiers with different parameter combinations can also be included in the ensemble; however, we constructed the ensemble classifier such that no candidate classifier is represented more than once in the candidate set. Also, hyperparameters of the candidate classifiers can be tuned, but we have chosen to use mostly default parameters of the candidate parameters. In the case where the default value of a parameter is not used, the value was chosen based on our experience in the analysis of metagenomic data. Nonetheless, since the default hyperparameters in some machine learning libraries may not be optimized for the classification problem at hand, we encourage analysts to consider tuning such parameters during analysis.

Furthermore, to evaluate the performance of the techniques discussed in Sects. 3.1 and 3.4 for feature selection and to overcome class imbalance, respectively, we will apply these methods to the species abundance table obtained from the Kraken2–Bracken system. The construction of the ensemble classifier can easily be modified to accommodate the implementation of these techniques.

4 Results

Here, we present results for the analysis described in Sect. 3.6. First, we describe the results obtained from the analysis of the species abundance tables obtained after taxonomic profiling was performed with MetaPhlAn2 (MP), Kraken2–Bracken (KB), and Kaiju (KJ), respectively. For each abundance table, further downstream pre-processing as discussed in Sect. 3.1 was carried out, and we obtained 1029, 4770, and 25,750 taxa for MP, KB, and KJ data, respectively. We performed a 10-fold split of the abundance data into 80% training and 20% test sets. For each split, we ensured each class was represented by at least three samples in both the training and test sets. The classification analysis was conducted by training the classifiers mentioned in Sect. 3.6 on the training set, while the test set was used to evaluate the performance of the models.

We used a consistent framework for the analysis of the respective abundance tables, that is, a pre-specified set of candidate classifiers and classifier parameters, performance measures, and resampling techniques were consistently employed across the analysis for each abundance table. However, we excluded the RPart classifier from the set of candidate classifiers when analyzing the KJ data; the classifier could not handle the vast number of features in this particular training set. Also, for the construction of the ensemble classifier, the number of bootstrap samples to be drawn, B , was set to be 50, while Kappa, multiclass G-mean, and MAUC were the performance measures used for performing weighted rank aggregation.

For the analysis of the abundance tables obtained from the respective taxonomic profiling tools, Table 2 shows the mean performance measures for each classification algorithm. Based on the results from all performance measures, and across the analysis for each profiling tool, the ensemble classifier yields classification results that are as good as the best candidate classifier. Furthermore, the candidate classifiers perform differently for each abundance data. For instance, based on the Kappa statistics, the MLP, PLS+RF, RF, and XGB were the best performing candidate classifiers for the analysis of the KB and KJ data, while the RF and XGB gave the most promising results for the analysis of the MP data. These classifiers proved to be the most competitive in the set of candidate classifiers; hence, the ensemble of classifiers across the analysis for each data set was mostly dominated by the MLP, PLS+RF, RF, and XGB classifiers. For each sub-table in Table 2, the last column shows the number of times each candidate classifier was the best performing local classifier in 500 instances (10 replications with 50 bootstrap iterations each).

Furthermore, the SVM with a radial basis kernel and the RPart classifiers yield moderate classification performance. Classifiers trained with integrated PLS terms performed better than classifiers with PCA terms; we observed that the PCA+RF classifier yields the poorest classification results among all candidate classifiers. Also, the PLS+RF classifier performed better than its RF counterpart for the analysis of the KB data, and the two classifiers have closely related results for the analysis of the KJ data, while the RF outperforms the PLS+RF classifier for the analysis of the MP data. In general, the trained classifiers yielded better performance results for the KB and KJ data than for the MP data.

For the second phase of our analysis, we sought to investigate the impact of both dimension reduction and techniques for handling class imbalance on the classification performance of the classifiers. In this regard, we have applied these methods solely for the analysis of the KB data. For each application, we follow a similar design of the analysis presented in the first paragraph of this section. For the weighted classifiers, class weights were computed as $w_c = 1/n_c$, where n_c is the number of samples in class c . While for over-sampling, the Gauss Noise (introduces Gaussian noise for the generation of synthetic samples) [39] over-sampling procedure was implemented. The HFE described in Sect. 3.1 was employed for dimension reduction. Table 3 shows the mean performance measures

Table 2 The mean performance measures (G-mean, Kappa, and MAUC) for a set of candidate classifiers and the ensemble classifier. The classifiers were trained on the species abundance data obtained after taxonomic profiling was performed with MetaPhlAn2, Kraken2–Bracken, and Kaiju

Classifier	MetaPhlAn2				Kraken2–Bracken				Kaiju			
	G-mean	Kappa	MAUC	Count	G-mean	Kappa	MAUC	Count	G-mean	Kappa	MAUC	Count
Ensemble	0.73	0.69	0.81	–	0.78	0.87	0.91	–	0.91	0.91	0.94	–
MLP	0.36	0.57	0.77	7	0.77	0.84	0.91	135	0.61	0.86	0.91	96
PCA+RF	0.00	0.02	0.64	0	0.00	0.03	0.66	0	0.00	0.08	0.66	0
PLS+AdaBoost	0.14	0.42	0.71	1	0.71	0.79	0.86	6	0.77	0.85	0.90	1
PLS+RF	0.28	0.55	0.75	10	0.77	0.87	0.91	212	0.87	0.88	0.92	51
PLS+RPart	0.07	0.31	0.69	0	0.46	0.52	0.74	0	0.37	0.58	0.77	0
PLS+XGB	0.17	0.54	0.75	5	0.72	0.81	0.88	0	0.67	0.84	0.89	0
RF	0.73	0.69	0.81	403	0.42	0.84	0.90	31	0.77	0.87	0.92	63
RPart	0.30	0.38	0.71	2	0.23	0.54	0.77	0				
SVM	0.10	0.43	0.73	0	0.65	0.74	0.86	0	0.48	0.67	0.83	0
XGB	0.44	0.65	0.80	72	0.87	0.88	0.92	116	0.90	0.90	0.94	289

Table 3 The mean performance measures for a set of candidate classifiers and the ensemble classifier. The classifiers were trained with a full feature space and a reduced feature space for the species abundance data obtained after taxonomic profiling was performed with Kraken2–Bracken. Classification results for the feature-reduced space obtained using HFE are shown in parentheses

Classifier	Standard			Weighted			Over-sampling		
	G-mean	Kappa	MAUC	G-mean	Kappa	MAUC	G-mean	Kappa	MAUC
Ensemble	0.78 (0.52)	0.87 (0.85)	0.91 (0.92)	0.87 (0.62)	0.87 (0.86)	0.92 (0.92)	0.86 (0.81)	0.87 (0.82)	0.93 (0.91)
MLP	0.77 (0.37)	0.84 (0.73)	0.91 (0.86)	0.64 (0.37)	0.8 (0.71)	0.89 (0.86)	0.81 (0.64)	0.8 (0.69)	0.9 (0.84)
PCA+RF	0 (0.00)	0.03 (0.04)	0.66 (0.65)	0 (0.00)	0.04 (0.04)	0.66 (0.64)	0 (0.00)	0.03 (0.06)	0.66 (0.67)
PLS+RF	0.77 (0.21)	0.87 (0.36)	0.91 (0.68)	0.87 (0.38)	0.87 (0.75)	0.92 (0.86)	0.85 (0.56)	0.86 (0.71)	0.92 (0.85)
PLS+RPart	0.46 (0.57)	0.52 (0.70)	0.74 (0.82)	0.42 (0.03)	0.51 (0.31)	0.76 (0.71)	0.3 (0.13)	0.49 (0.30)	0.75 (0.71)
PLS+XGB	0.72 (0.35)	0.81 (0.85)	0.88 (0.92)	0.72 (0.35)	0.81 (0.70)	0.89 (0.84)	0.78 (0.46)	0.8 (0.67)	0.88 (0.83)
RF	0.42 (0.47)	0.84 (0.53)	0.9 (0.76)	0.51 (0.62)	0.84 (0.86)	0.92 (0.92)	0.63 (0.81)	0.81 (0.83)	0.91 (0.91)
RPart	0.23 (0.39)	0.54 (0.36)	0.77 (0.69)	0.58 (0.39)	0.56 (0.55)	0.79 (0.79)	0.45 (0.36)	0.54 (0.51)	0.78 (0.76)
SVM	0.65 (0.76)	0.74 (0.85)	0.86 (0.92)	0.1 (0.10)	0 (0.00)	0.5 (0.50)	0.63 (0.48)	0.72 (0.41)	0.86 (0.72)
XGB	0.87 (0.52)	0.88 (0.85)	0.92 (0.92)	0.85 (0.84)	0.86 (0.84)	0.91 (0.90)	0.83 (0.81)	0.84 (0.82)	0.91 (0.90)

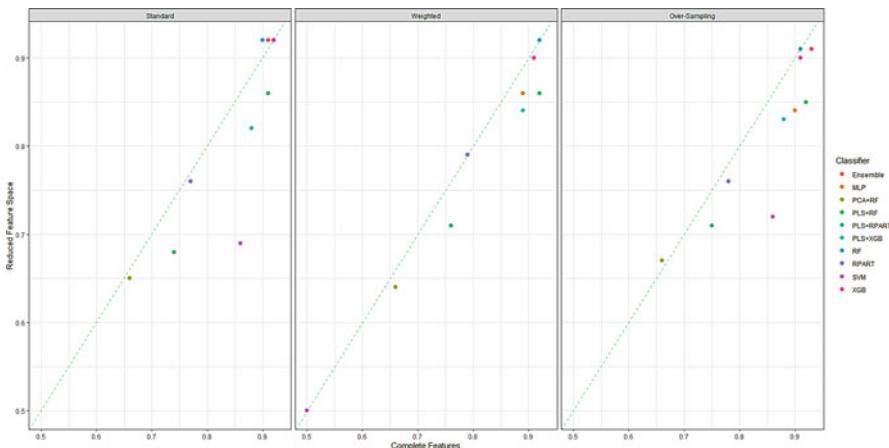


Fig. 3 Mean multiclass AUC measures for ten standard classifiers, and an ensemble classifier comprising of the standard classifiers. These classifiers were trained with the species abundance table obtained after taxonomic profiling was done with the Kraken2–Bracken system. The training data with a set of complete features comprise 4770 taxa that were obtained after downstream pre-processing, while the data with a reduced feature space comprise 796 taxa, on average

for the set of candidate classifiers and the ensemble classifier, and the classification results for the data with a reduced feature space are shown in parentheses. First, by contrasting the classification performance for the HFE and non-HFE data across the three different techniques shown in the sub-tables of Table 3, notice that there is little or no improvement in classification results for the feature-reduced data. For most of the results reported, the classifiers performed slightly better on the non-HFE data.

Also, for comparison of classification results across the methods used to address the problem of class imbalance and the standard classifiers, we find that there is no substantial improvement in classification performance. Figure 3 shows the mean multiclass AUC scores for the standard classifiers as well as the classifiers trained with class weights and oversampled data. The classifiers are trained on both the non-HFE and HFE data. For each classifier, the multiclass AUCs reported for all three approaches are very similar. This finding is consistent with the description that the class weighting and over-sampling techniques do not improve the overall fit of the models.

We further investigated the performance of the classifiers when predicting the known class labels in the primary data. The classifiers had a varied performance for prediction of the sample origins. Figure 4 shows a boxplot of the positive predictive values (PPV) based on the classification results from the standard ensemble classifier (i.e., class weighting and over-sampling procedure were not applied) trained on a full feature space. The PPV results described here were

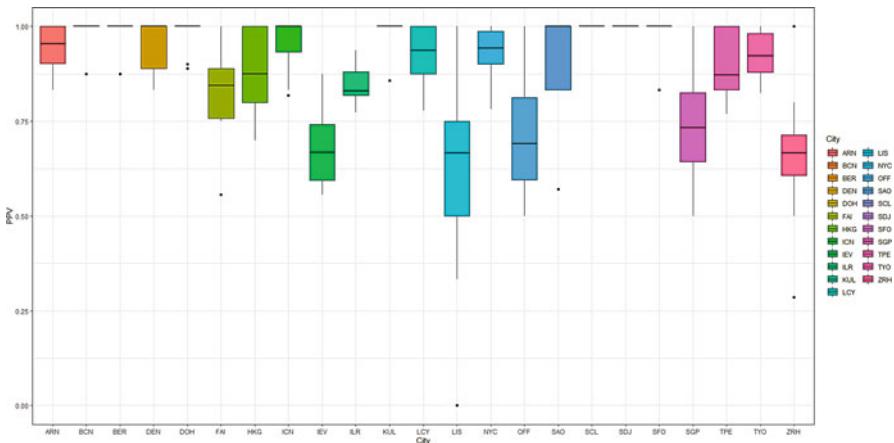


Fig. 4 Boxplot showing the positive predictive value for all cities represented in the training data. The results are based on predictions from a standard ensemble classifier that was trained on the full feature space of the species abundance data obtained after taxonomic profiling was performed with Kraken2–Bracken

obtained for the analysis of KB data discussed in the first paragraph of this section. The classifier yields near perfect prediction for samples obtained from *Barcelona*, *Berlin*, *Denver*, *Doha*, *Kuala Lumpur*, *Offa*, *Santiago*, *Sendai*, *San Francisco*, and *Tokyo*. The average PPV for prediction of these sample origins was at least 95%. In contrast, the ensemble classifier does not yield good classification performance for the prediction of samples that originated from *Kiev*, *Lisbon*, *Offa*, and *Singapore*. The average PPV for these cities ranges from 60% to 74%. The poor performance of the classifier in predicting certain cities will negatively impact the overall classification performance of the classifier. Thus, it is worthwhile to investigate the reasons for the poor predictive ability of the classifier for these cities. For instance, we observed that the classifier had trouble discriminating between *Kiev* and *Zurich*. Certain factors could influence the sub-par ability of the classifier in discriminating between cities. The proximity of source cities is an obvious factor. Naturally, we can expect the classifiers to misclassify cities in close proximity to one another. For instance, *Offa* and *Ilorin* are geographically close, and the classifier, in several cases, misclassified *Offa* as the *Ilorin*.

The boxplots in Fig. 5 show some of the top microbial species that were found to be differentially abundant across various cities. The left panel of Fig. 5 shows the feature importance plot of the top 20 species from RF classifier in the ensemble. Variable importance plot consists of many species belonging to genus *Bradyrhizobium* that is a soil bacteria and is also found in the roots and stems of plants [27]. *Pseudomonas.sp..CC6.YY.74* species belongs to genus *Pseudomonas* that is a common genus of bacteria that resides on moist surfaces, soil, and water [42].

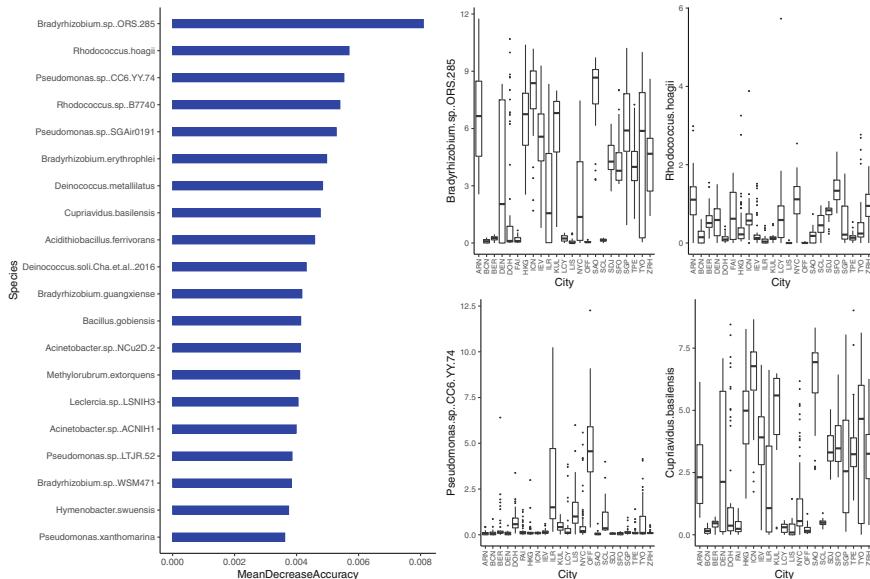


Fig. 5 Species importance for RF classifier in the ensemble (left). Boxplots of species abundances for 4 among the top 10 important species (right)

5 Discussion

We have presented a practical workflow for the analysis of microbiome data that are based on samples that are usually collected from the different body and environmental sites. This workflow was partitioned into two sections—pre-processing of raw WGS data and downstream analysis. For the raw WGS data pre-processing of the microbiome data, we constructed a standard pipeline using a variety of bioinformatics tools for quality control and taxonomic profiling. The taxonomic profiling involves classifying sequence reads to taxonomic labels and estimation of species abundance, and this was performed with three widely used profiling tools, namely, MetaPhlAn2, Kraken2–Bracken, and Kaiju. At the termination of the bioinformatics pipeline, we obtain species abundance tables from each of the respective profiling tools, and these abundance tables were passed into the downstream analysis.

The downstream analysis of the data comprised fitting supervised learning models for the classification of the species abundance of the samples to known class labels. We have evaluated several machine learning approaches to the metagenomics-based classification of sample origins. For this purpose, we adopted a robust ensemble classifier that uses species-level abundance as features, a user-specific set of supervised learning models as candidate classifiers, and user-defined performance metrics for model evaluation. The ensemble classifier is an adaptive

classification algorithm that can perform well on different data structures. This classifier utilizes performance on OOB samples to guard against overfitting. The ensemble classifier gives classification performance better or as good as the best performing candidate classifier in its ensemble.

Across many metagenomics studies, we noticed a great deal of variation in classification results presented by different researchers working in this area. One natural explanation for this variation in results stems from the bioinformatics and data generation procedures employed in these studies. Since standard classification models will perform differently when trained on different data structures, restricting the classification problem to a single classifier may not be a practical approach. For a given classification problem, the analyst is expected to try out a variety of classifiers, judging each one according to a set of user-defined performance metrics. In this sense, the analyst will likely begin their exploration with simple models before trying out more complex models. With the application of the ensemble classifier described here, the analyst can automate the process and achieve a near optimal performance.

In this chapter, we have trained the ensemble classifier with only the classifiers discussed in Sect. 3.2. However, the ensemble need not be restricted to these models but could include any reasonable user-specified classifier. For instance, we notice that the XGBoost classifier that is popular among competing teams solving data science problems has been rarely used in the analysis of metagenomics data. Results from classification performance presented in this chapter showed that the XGBoost performs almost as well as the RF classifier. Therefore, in a future analysis of these types of data, we may choose to include XGBoost in our ensemble.

The best classification results for the prediction of source cities were obtained when the classifiers were trained on the full data set rather than on the feature-reduced version. This explains the complex nature of metagenomics data where a plethora of taxa are needed to characterize the variation among sample origins; hence, building a model with only a subset of these taxa may not sufficiently explain such variations.

In addition to fitting an ensemble of classifiers, we also highlight other techniques that may improve the classification of metagenomics data. Since most machine learning models tend to lean toward predicting the majority classes over the minority classes, balancing the class frequencies of samples in the training data is an ideal method to incorporate in the analytical pipeline. The application of an optimal minority over-sampling scheme and class weighting in the training of the classifiers only marginally impacted the performance of the classifiers presented in this chapter. These techniques can easily be incorporated while constructing the ensemble classifier. We notice that training the classifier with class weights is computationally more efficient than utilizing an over-sampling scheme.

An obvious drawback of the ensemble classifier is that it is computationally intensive. It would take more time to train an ensemble classifier than it would for a stand-alone classifier. The computing times of the ensemble classifier are mainly impacted by the number of bootstrap samples that the individual classifiers are

trained on, the number and complexity of user-specified candidate classifiers, and the performance measures that are used to compute the weighted rank aggregation. However, the computing times can be appreciably reduced if the ensemble classifier is trained using parallel computational approaches on a computing cluster. When we selected 10 candidate classifiers (i.e., the candidate classifiers presented in Sect. 4), three performance measures (namely, Cohen’s Kappa coefficient, multiclass G-mean, and AUC) for computing weighted rank aggregation, and 50 bootstrap samples for the construction of the ensemble classifier, the construction procedure took an average time of 9.47 h (wall-clock time). This procedure was done on a University computing cluster for which 12 CPU cores and 40GB of memory were allocated to the job.

The downstream classification analysis presented here can be extended in two different directions. Each of these extensions requires the knowledge of additional information besides the microbiome data—such information are often present in the form of geographic location of the training cities or the weather information in both training and test cities and so on. In the former case, we can build a potentially improved classifier that effectively utilizes a larger collection of features. In the later situation, one may be able to predict the city of origin in a bigger list than what was provided in the training data. These extensions may be pursued elsewhere.

6 Data Acknowledgement

All analyses presented in this chapter are based on the raw WGS metagenomics data provided as part of the 2020 CAMDA Metagenomic Geolocation Challenge. The primary data along with other supplementary data is publicly available on the challenge’s [website](#). We participated in this challenge and presented our classification results at the [2020 Intelligent Systems for Molecular Biology](#) (ISMB) conference. An extensive report on the results from our analysis will be published in the conference proceedings.

7 Code Availability

Bash scripts for each procedure performed in the bioinformatics pipeline and R scripts for building an ensemble of standard classifiers are available at https://github.com/samuelanyaso/metagenomic_data_analysis. The sample code below shows a standard interface to analyze an abundance matrix. The code calls the `ensemble.R` script for training an ensemble classifier, predicts test cases, and evaluates the performance of the ensemble classifier along with other candidate classifiers in the ensemble.

```

WD <- "/path/to/data/and/source/scripts"
setwd(WD)
df <- read.delim("abundanceTable.txt", header = TRUE, sep = "\t",
                 dec = ".")
df$class <- factor(df$class) # class labels
## Begin training Models
num.class <- length(levels(df$class))
idx <- 1:nrow(df) # row indices
## loads the ensemble function
source("ensemble.R")

Result1 <- list()
Result2 <- list()
bestAlg <- list()
confMat <- list()
reps <- 10 # number of replications
set.seed(2021)
for(r in 1:reps){
  repeat{
    ## repeat partitioning of the data into train and test set until
    ## all classes are present in both test and train set
    inTraining <- createDataPartition(df$class,p = 0.9,list = FALSE)
    shuf <- sample(inTraining[,1],replace = FALSE) # train set
    shufT <- sample(idx[which(!idx %in% inTraining[,1])],
                    replace = FALSE) # test set
    # partitions the dataset
    dat.train <- df[shuf,]
    dat.test <- df[shufT,]
    if(all(table(dat.train$class) >= 1) & all(table(dat.test$class)
                                                >= 1)){
      break
    }
  }
  ## Train set
  y <- dat.train$class
  y <- as.factor(as.numeric(y)-1) # Factor levels should begin from 0
  x <- data.matrix(dat.train[,!(names(dat.train) %in% c("class"))])
  ## Test set
  yTest <- dat.test$class
  yTest <- as.factor(as.numeric(yTest)-1) # Factor levels should
  begin from 0
  xTest <- data.matrix(dat.test[,!(names(dat.test) %in% c("class"))])
  cat("Started Replication: ",r," of ",reps,"\n")
  ens <- ensembleClassifier(x, y, M=50, ncomp=30,
                            train = dat.train, test = dat.test,
                            algorithms=c("svm","rang","pls_rf",
                                         "pca_rf","rpart", "pls_rpart",
                                         "xgb","pls_xgb","mlp"),
                            levsChar =as.character(levels(dat.train$class)))
  # the names of the best local classifiers
  bestAlg[[r]] <- ens$bestAlg
  ## predict using the test data
  pred <- predictEns(ens, xTest, yTest, test = dat.test,
                     dlEnsPath = "dl_ens_time.h5",
                     dlIndPath = "dl_ind_time.h5")
  # Saves the results
}

```

```

Result1[[r]] <- pred$ensemblePerf
Result2[[r]] <- pred$indPerf
## predicted class
yPred <- pred$yhat
## confusion matrix
confMat[[r]] <- caret::confusionMatrix(yPred,yTest)
# displays the truth and predictions for each of the "best" algorithms
dfPred <- data.frame(truth=yTest, ensemble=yPred, pred$pred)
dfPred <- as.list(dfPred)
# convert numeric factors to character factors
dfPred <- lapply(dfPred,function(x)
  as.character(num2charFac(x,char.levs =
    as.character(levels(dat.train$class)))))
names(dfPred) <- c("truth","ensemble",ens$bestAlg)
dfPred <- as.data.frame(dfPred)
cat("Predictions for the best individual
models for iteration: ",r," of ",reps,"\n ")
print(dfPred)
cat("Completed Replication: ",r," of ",reps,"\n ")
}
# save performance results
saveRDS(Result1,"ensClassifPerf.RDS")
saveRDS(Result2,"indClassifPerf.RDS")
saveRDS(bestAlg,"bestAlg.RDS")
saveRDS(confMat,"confMat.RDS")
warnings()

```

Acknowledgments This work was partially supported by the National Science Foundation under Award DMS-1461948 to SG.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Mol. Biol.* **215**(3), 403–410 (1990)
2. Andrews, S.: FastQC. <http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>. Accessed: 2020-08-28
3. Bharti, R., Grimm, D.G.: Current challenges and best-practice protocols for microbiome analysis. *Brief. Bioinform.* **22**(1), 178–193 (2019). <https://doi.org/10.1093/bib/bbz155>
4. Bolger, A.M., Lohse, M., Usadel, B.: Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics* **30**(15), 2114–2120 (2014)
5. Branco, P., Ribeiro, R.P., Torgo, L.: UBL: an R package for utility-based learning. Preprint (2016). arXiv:1604.08079
6. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
7. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. CRC Press, Boca Raton, FL (1984)
8. Breitwieser, F.P., Lu, J., Salzberg, S.L.: A review of methods and databases for metagenomic classification and assembly. *Brief. Bioinform.* **20**(4), 1125–1136 (2019)
9. Buchfink, B., Xie, C., Huson, D.H.: Fast and sensitive protein alignment using DIAMOND. *Nature Methods* **12**(1), 59–60 (2015)

10. Casimiro-Soriguer, C.S., Loucera, C., Perez Florido, J., López-López, D., Dopazo, J.: Antibiotic resistance and metabolic profiles as functional biomarkers that accurately predict the geographic origin of city metagenomics sample. *Biology Direct* **14**, 15 (2019)
11. Chase, J., Fouquier, J., Zare, M., Sonderegger, D.L., Knight, R., Kelley, S.T., Siegel, J., Caporaso, J.G.: Geography and location are the primary drivers of office microbiome composition. *mSystems* **1**(2), e00022-16 (2016)
12. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. Statistics Department of University of California at Berkeley, Berkeley. Technical Report 666 (2004)
13. Chen, T., Guestrin, C.: XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA, 2016), KDD '16, pp. 785–794. Association for Computing Machinery (2016)
14. Claesson, M.J., Clooney, A.G., O'Toole, P.W.: A clinician's guide to microbiome analysis. *Nat. Rev. Gastroenterol. Hepatol.* **14**(10), 585–595 (2017)
15. Clarke, T.H., Gomez, A., Singh, H., Nelson, K.E., Brinkac, L.M.: Integrating the microbiome as a resource in the forensics toolkit. *Forensic Sci. Int. Genet.* **30**, 141–147 (2017)
16. Cohen, J.: A coefficient of agreement for nominal scales. *Educ. Psychol. Meas.* **20**(1), 37–46 (1960)
17. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* **20**(3), 273–297 (1995)
18. Datta, S., Pihur, V., Datta, S.: An adaptive optimal ensemble classifier via bagging and rank aggregation with applications to high dimensional data. *BMC Bioinformatics* **11**, 427 (2010)
19. Ditzler, G., Morrison, J.C., Lan, Y., Rosen, G.L.: Fuzzy: feature subset selection for metagenomics. *BMC Bioinformatics* **16**(1), 358 (2015)
20. Duvallet, C., Gibbons, S.M., Gurry, T., Irizarry, R.A., Alm, E.J.: Meta-analysis of gut microbiome studies identifies disease-specific and shared responses. *Nature Communications* **8**(1), 1–10 (2017)
21. Ewels, P., Magnusson, M., Lundin, S., Käller, M.: MultiQC: Summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **32**(19), 3047–3048 (2016)
22. Ewing, B., Green, P.: Base-calling of automated sequencer traces using Phred. II. Error probabilities. *Genome Res.* **8**(3), 186–194 (1998)
23. Freitas, T.A.K., Li, P.-E., Scholz, M.B., Chain, P.S.G.: Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. *Nucleic Acids Res.* **43**(10), e69 (2015)
24. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.* **55**(1), 119–139 (1997)
25. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.*, 1189–1232 (2001)
26. Gilbert, J.A., Jansson, J.K., Knight, R.: The earth microbiome project: successes and aspirations. *BMC Biology* **12**(1), 69 (2014)
27. Giraud, E., Xu, L., Chaintreuil, C., Gargani, D., Gully, D., Sadowsky, M.J.: Photosynthetic *Bradyrhizobium* sp. strain ORS285 is capable of forming nitrogen-fixing root nodules on soybeans (*glycine max*). *Appl. Environ. Microbiol.* **79**(7), 2459–2462 (2013)
28. Grice, E.A., Segre, J.A.: The human microbiome: Our second genome. *Annu. Rev. Genomics Hum. Genet.* **13**(1), 151–170 (2012). PMID: 22703178
29. Hand, D.J.: Breast cancer diagnosis from proteomic mass spectrometry data: A comparative evaluation. *Stat. Appl. Genet. Mol. Biol.* **7**(2), Article 15 (2008)
30. Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning* **45**(2), 171–186 (2001)
31. Harris, Z.N., Dhungel, Eliza Mosior, M., Ahn, T.-H.: Massive metagenomic data analysis using abundance-based machine learning. *Biology Direct* **14**(12), Article 12 (2019)
32. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer Series in Statistics. Springer New York, New York, NY, USA (2001)
33. Huttenhower, C., et al.: Structure, function and diversity of the healthy human microbiome. *Nature* **486**(7402), 207–214 (2012)

34. Joshi, M.V., Kumar, V., Agarwal, R.C.: Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In: Proceedings 2001 IEEE International Conference on Data Mining, San Jose, CA, USA pp. 257–264. IEEE (2001)
35. Knights, D., Costello, E.K., Knight, R.: Supervised classification of human microbiota. *FEMS Microbiol. Rev.* **35**(2), 343–359 (2011)
36. Kovács, G.: An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets. *Appl. Soft Comput.* **83**, 105662 (2019a)
37. Kovács, G.: Smote-variants: a python implementation of 85 minority oversampling techniques. *Neurocomputing* **366**, 352–354 (2019b)
38. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with Bowtie 2. *Nature Methods* **9**(4), 357–359 (2012)
39. Lee, S.S.: Regularization in skewed binary classification. *Computational Statistics* **14**, 277–292 (1999)
40. Li, H.: Microbiome, metagenomics, and high-dimensional compositional data analysis. *Annu. Rev. Stat. Appl.* **2**, 73–94 (2015)
41. Lin, W., Shi, P., Feng, R., Li, H.: Variable selection in regression with compositional covariates. *Biometrika* **101**(4), 785–797 (2014)
42. Lin, X., Zhang, Z., Zhang, L., Li, X.: Complete genome sequence of a denitrifying bacterium, *Pseudomonas* sp. CC6-YY-74, isolated from Arctic Ocean sediment. *Marine Genomics* **35**, 47–49 (2017)
43. Lindgreen, S., Adair, K.L., Gardner, P.P.: An evaluation of the accuracy and speed of metagenome analysis tools. *Scientific Reports* **6**, 19233 (2016)
44. Lu, J., Breitwieser, F.P., Thielen, P., Salzberg, S.L.: Bracken: estimating species abundance in metagenomics data. *PeerJ Comput. Sci.* **3**, e104 (2017)
45. Marchesi, J.R., Ravel, J.: The vocabulary of microbiome research: a proposal. *Microbiome* **3**(1), 31 (2015)
46. Martin, M.: Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet J.* **17**(1), 10–12 (2011)
47. Mason, C., Hirschberg, D., Consortium, T.M.I.: The metagenomics and metadesign of the subways and urban biomes (MetaSub) International Consortium inaugural meeting report. *Microbiome* **4**(1), 24 (2016)
48. McIntyre, A.B., Ounit, R., Afshinnekoo, E., Prill, R.J., Hénaff, E., Alexander, N., Minot, S.S., Danko, D., Foox, J., Ahsanuddin, S., et al.: Comprehensive benchmarking and ensemble approaches for metagenomic classifiers. *Genome Biology* **18**(1), 182 (2017)
49. McIver, L.J., Abu-Ali, G., Franzosa, E.A., Schwager, R., Morgan, X.C., Waldron, L., Segata, N., Huttenhower, C.: bioBakery: a meta-omic analysis environment. *Bioinformatics* **34**(7), 1235–1237 (2017)
50. Menzel, P., Ng, K.L., Krogh, A.: Fast and sensitive taxonomic classification for metagenomics with Kaiju. *Nature Communications* **7**, 11257–11257 (2016)
51. Moitinho-Silva, L., Steinert, G., Nielsen, S., Hardoim, C.C., Wu, Y.-C., McCormack, G.P., López-Legentil, S., Marchant, R., Webster, N., Thomas, T., et al.: Predicting the HMA-LMA status in marine sponges by machine learning. *Front. Microbiol.* **8**, 752 (2017)
52. Oudah, M., Henschel, A.: Taxonomy-aware feature engineering for microbiome classification. *BMC Bioinformatics* **19**, 227 (2018)
53. Ounit, R., Wanamaker, S., Close, T.J., Lonardi, S.: Clark: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics* **16**(1), 236 (2015)
54. 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**(7), e1004977 (2016)
55. Paulson, J.N., Pop, M., Bravo, H. C.: metagenomeSeq: Statistical analysis for sparse high-throughput sequencing Bioconductor package **1**(0), 191 (2013)
56. Pihur, V., Datta, S., Datta, S.: Weighted rank aggregation of cluster validation measures: a Monte Carlo cross-entropy approach. *Bioinformatics* **23**(13), 1607–1615 (2007)

57. Ranjan, R., Rani, A., Metwally, A., McGee, H.S., Perkins, D.L.: Analysis of the microbiome: Advantages of whole genome shotgun versus 16s amplicon sequencing. *Biochem. Biophys. Res. Commun.* **469**(4), 967–977 (2016)
58. Schmieder, R., Edwards, R.: Insights into antibiotic resistance through metagenomic approaches. *Future Microbiology* **7**, 73–89 (2012)
59. Segata, N., Waldron, L., Ballarini, A., Narasimhan, V., Jousson, O., Huttenhower, C.: Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature Methods* **9**, 811–814 (2012)
60. Shi, P., Zhang, A., Li, H., et al.: Regression analysis for microbiome compositional data. *Ann. Appl. Stat.* **10**(2), 1019–1040 (2016)
61. Singh, R.K., Chang, H.-W., Yan, D., Lee, K.M., Ucmak, D., Wong, K., Abrouk, M., Farahnik, B., Nakamura, M., Zhu, T.H., Bhutani, T., Liao, W.: Influence of diet on the gut microbiome and implications for human health. *J. Transl. Med.* **15**(1), 73 (2017)
62. Sun, Y., Kamel, M.S., Wang, Y.: Boosting for learning multiple classes with imbalanced class distribution. In: Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China. pp. 592–602. IEEE (2006)
63. Sunagawa, S., Mende, D.R., Zeller, G., Izquierdo-Carrasco, F., Berger, S.A., Kultima, J.R., Coelho, L.P., Arumugam, M., Tap, J., Nielsen, H.B., Rasmussen, S., Brunak, S., Pedersen, O., Guarner, F., de Vos, W.M., Wang, J., Li, J., Doré, J., Ehrlich, S.D., Stamatakis, A., Bork, P.: Metagenomic species profiling using universal phylogenetic marker genes. *Nature Methods* **10**(12), 1196–1199 (2013)
64. Tibshirani, R.: Regression shrinkage and selection via the Lasso. *J. Roy. Stat. Soc. B (Methodol.)* **58**(1), 267–288 (1996)
65. Turnbaugh, P.J., Ley, R.E., Hamady, M., Fraser-Liggett, C.M., Knight, R., Gordon, J.I.: The human microbiome project. *Nature* **449**(7164), 804–810 (2007)
66. Wade, W.: The oral microbiome in health and disease. *Pharmacological Research* **69**(1), 137–143 (2013). Copyright 2012 Elsevier Ltd. All rights reserved
67. Wang, W.-L., Xu, S.-Y., Ren, Z.-G., Tao, L., Jiang, J.-W., Zheng, S.-S.: Application of metagenomics in the human gut microbiome. *World J. Gastroenterol.* **21**(3), 803–814 (2015)
68. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology* **15**, Article R46 (2014)
69. Wood, D.E., Lu, J., Langmead, B.: Improved metagenomic analysis with Kraken 2. *Genome Biology* **20**(1), 257 (2019)
70. Ye, S.H., Siddle, K.J., Park, D., Sabeti, P.C.: Benchmarking metagenomics tools for taxonomic classification. *Cell* **178**, 779–794 (2019)
71. Zhou, Q., Su, X., Ning, K.: Assessment of quality control approaches for metagenomic data analysis. *Scientific Reports* **4**(1), 6957 (2014)
72. Zhou, Y.-H., Gallins, P.: A review and tutorial of machine learning methods for microbiome host trait prediction. *Front. Genet.* **10**, 579 (2019)
73. Zou, H., Hastie, T.: Regularization and variable selection via the ElasticNet. *J. Roy. Stat. Soc. B (Stat. Methodol.)* **67**(2), 301–320 (2005)

Part II

**Exploratory Analyses of Microbial
Communities**

Statistical Methods for Pairwise Comparison of Metagenomic Samples



Kai Song and Fengzhu Sun

1 Introduction

Microbes are widely distributed in various environments on the earth's surface and different parts of the human body, such as human gut, skin, and oral cavity. The number of microbial cells in different parts of human body is estimated to be 10 times more than the number of human cells. Traditional microbial research relies heavily on laboratory culture. However, only a small number of microorganisms can be successfully cultured in the laboratory making culture-based methods not widely applicable in many environments. Metagenomic sequencing technologies provide a powerful approach to study the microorganisms directly from the environment. Two main types of approaches, namely marker gene such as 16S rRNA profiling and whole metagenome shotgun sequencing, are widely used in the field to investigate complex microbial communities.

The comparison of microbial communities is a highly important problem in ecological research. Many measures generally referred as beta diversity for the comparison of microbial communities have been developed. In this chapter, we critically review beta diversity measures based on either marker gene profiling data or metagenomic shotgun sequencing data. Beta diversity measures provide quantitative measurements of differences between two microbial communities, laying the foundation for quantitative comparison of multiple samples. The beta

K. Song

School of Mathematics and Statistics, Qingdao University, Qingdao, China

e-mail: ksong@qdu.edu.cn

F. Sun (✉)

Department of Quantitative and Computational Biology, University of Southern California, Los Angeles, CA, USA

e-mail: fsun@usc.edu

diversity values between all pairs of samples as a dissimilarity matrix can be used to study group relationships of the microbial communities and to understand the relationships between microbial communities and environmental gradients such as temperature and geographical location.

The organization of this chapter is as follows. In Sect. 2, we provide classic measures for comparing microbial communities based on the relative abundance profiles of different operational taxonomic units (OTUs) including the Euclidean, Bray–Curtis, and Jaccard distances. These measures can be applied to both marker gene profiling and shotgun sequencing data. In Sect. 3, we review various relatively newly developed beta diversity measures based on the phylogenetic relationships of the different OTUs and possibly their abundance profiles including UniFrac, weighted UniFrac, and variance adjusted weighted UniFrac. These measures have been mostly used for the analyses of marker gene profiling data. In Sect. 4, we review recently developed alignment-free methods for the comparison of microbial communities based on word pattern (k -mer) occurrences including Composition Vector Tree (*CVTree*), d_2^S , and d_2^* . In Sect. 5, we present some practical examples using these measures to compare microbial communities. The chapter concludes with a discussion of the advantages and disadvantages of the different measures and future directions for research on the comparison of microbial communities.

2 Microbial Community Comparison Methods Based on OTU Abundance Data

For 16S rRNA marker gene profiling data, the reads are either clustered into different groups or mapped to existing 16S rRNA databases to obtain the absolute abundance of different OTUs in each sample. To remove the potential confounding effects of the number of reads on beta diversity, the absolute abundance is normalized by the total number of reads in each sample to obtain the relative abundance of each OTU in each sample. Suppose we have a matrix, $A = (A_{ij})_{N \times M}$, where i represents the sample and j represents the OTU, and A_{ij} is the relative abundance of the j th OTU in the i th sample. N and M are the total number of samples and OTUs, respectively. Both quantitative and qualitative measures have been used to compare microbial communities based on the relative OTU abundance data. The quantitative measures, such as the Bray–Curtis dissimilarity, Canberra dissimilarity, and Euclidean distance, consider the abundance of each OTU in the community. The qualitative measures, such as Jaccard index and Dice coefficient, consider only the presence/absence of each OTU in the communities, regardless of their abundance. The commonly used quantitative measures for the comparison of two samples a and b are as follows.

The Bray–Curtis dissimilarity,

$$D_{ab} = \frac{\sum_{j=1}^M |A_{aj} - A_{bj}|}{2}. \quad (1)$$

The Canberra dissimilarity,

$$D_{ab} = \frac{1}{Num_{a \cup b}} \sum_{j \in a \cup b} \frac{|A_{aj} - A_{bj}|}{A_{aj} + A_{bj}}, \quad (2)$$

where $Num_{a \cup b}$ is the number of OTUs (columns) that are present in either sample a or sample b , that is, with abundance greater than 0. The summation is over all the OTUs that are present in at least one of the samples.

The Euclidean distance,

$$D_{ab} = \sqrt{\sum_{j=1}^M (A_{aj} - A_{bj})^2}. \quad (3)$$

The Chord distance,

$$D_{ab} = \sqrt{\sum_{j=1}^M \left(\frac{A_{aj}}{\sqrt{\sum_{j=1}^M A_{aj}^2}} - \frac{A_{bj}}{\sqrt{\sum_{j=1}^M A_{bj}^2}} \right)^2}. \quad (4)$$

The Gower distance,

$$D_{ab} = \sum_{j=1}^M \frac{|A_{aj} - A_{bj}|}{(\max_i A_{ij} - \min_i A_{ij})}, \quad (5)$$

where the maximum and minimum are taken over all the samples.

The Hellinger distance,

$$D_{ab} = \sqrt{\sum_{j=1}^M \left(\sqrt{A_{aj}} - \sqrt{A_{bj}} \right)^2}. \quad (6)$$

In addition, one minus the Pearson or Spearman correlation coefficient between A_a and A_b can be used to measure the distance between microbial communities a and b .

The qualitative measures for the comparison of two samples a and b consider only the presence/absence of OTUs and do not take their abundance levels into consideration. Let Num_a be the number of OTUs present in a sample a and $a \cap b$

be the set of OTUs present in both samples a and b . The commonly used qualitative measures include the following.

The Hamming dissimilarity,

$$D_{ab} = \text{Num}_a + \text{Num}_b - 2\text{Num}_{a \cap b}. \quad (7)$$

The Jaccard dissimilarity,

$$D_{ab} = 1 - \frac{\text{Num}_{a \cap b}}{\text{Num}_a + \text{Num}_b - \text{Num}_{a \cap b}}. \quad (8)$$

The Dice coefficient,

$$D_{ab} = 1 - \frac{2\text{Num}_{a \cap b}}{\text{Num}_a + \text{Num}_b}. \quad (9)$$

The Ochiai coefficient,

$$D_{ab} = 1 - \frac{\text{Num}_{a \cap b}}{\sqrt{\text{Num}_a \times \text{Num}_b}}. \quad (10)$$

The Lennon dissimilarity,

$$D_{ab} = 1 - \frac{\text{Num}_{a \cap b}}{\text{Num}_{a \cap b} + \min(\text{Num}_a - \text{Num}_{a \cap b}, \text{Num}_b - \text{Num}_{a \cap b})}. \quad (11)$$

More information about the OTU-based measures and their performances can be found from [8]. In their study, the authors used two real datasets and simulated 16S rRNA samples to evaluate the performances of 51 different OTU-based measures to reveal two different underlying relationships: environmental gradient and sample clustering. The two real datasets include microbial communities from different fingertips and keyboards and from soil with different pH values. One of the simulated samples used the unimodal abundance curves to mimic species relative abundance levels that were assumed to be affected by an environmental gradient. Different gradient locations along the curves were chosen to represent the abundance levels of the different species in these samples. The abundance levels for the species at the same location were normalized and used to generate the datasets. For the other simulated samples, the authors simulated clusters of metagenomic samples linked by a tree. At each level of the tree, the abundance vectors were randomized and renormalized to generate the abundance vector of each sample in the clusters for generating the datasets. It was shown that the Gower and χ^2 distances perform better than other measures in revealing the relationships among the samples along an environmental gradient even under low sequencing depth and the presence of noise in simulating abundance vectors. The Jaccard dissimilarity measure was shown to perform well in revealing the cluster relationships when such relationships were prominent, but not very well for subtle clusters. The χ^2

distance was shown to perform well in revealing environmental gradients, but only moderately well in revealing underlying clusters.

The advantage of the OTU-based measures is that they can be used flexibly for comparing different metagenomic samples from different individuals and environments. However, when the OTU composition of different microbial communities is highly heterogeneous, the OTUs of different communities are very different, which could affect the effectiveness of these measures. Many OTU-based measures are available, and their performances are highly different [8]. Therefore, it is important to choose an appropriate beta diversity measure for comparing microbial communities. In addition, none of these OTU-based methods can eliminate the arch effect (an arch configuration of samples shown in a two-dimensional plane simulated under a single environmental gradient). The performances of these methods were only compared using simple simulated datasets with only one underlying environmental gradient. However, multiple underlying environmental gradients may affect the composition of microbial communities in nature. The performances of these measures under complex environmental gradients need to be investigated in the future. The performances of these measures also depend on the sequencing depth of the microbial samples. Further research is needed to investigate the impacts of sequencing depth on the performances of the different OTU-based measures to uncover the relationships among microbial samples.

3 Microbial Community Comparison Measures Based on a Phylogenetic Tree

3.1 *The F_{ST} Statistic and Phylogenetic Test for Comparing Communities*

The OTU-based measures consider each OTU in microbial communities equally and ignore the evolutionary relationships among these OTUs. The phylogenetic-based measures are those taking evolutionary relationship among sequences into consideration when comparing microbial communities. Martin [17] developed the F_{ST} and the Phylogenetic (P) test statistics for comparing two microbial communities. The F_{ST} statistic measures the difference between two communities by comparing the sequence diversity of each community with the diversity of communities combined. The F_{ST} statistic is defined as

$$F_{ST} = \frac{\theta_T - \theta_W}{\theta_W}, \quad (12)$$

where θ_T is the sequence diversity of the combined community, and θ_W is the sequence diversity within each community averaged over all the communities being compared.

Various statistics can be used for estimating the sequence diversity θ . The average sequence divergence is the expected number of nucleotide differences between any pair of sequences chosen from a population. It is calculated as

$$\theta = \sum_{i=1}^k \sum_{j < i} p_i p_j d_{ij}, \quad (13)$$

where k is the number of distinct sequences, p_i is the frequency of the i th sequence, p_j is the frequency of the j th sequence, and d_{ij} is the number of differences between two sequences i and j . Then, the value of θ is divided by the length of the sequences compared to obtain the average nucleotide diversity that reflects the probability two randomly chosen sequences differ at a single base position. The population differentiation statistic, F_{ST} , can be calculated using a variety of different computer programs, such as Arlequin [4] and Variscan [24]. To evaluate the statistical significance of the observed value of F_{ST} , the sequences are randomly labeled in the microbial communities to generate the randomized F_{ST} distribution. The p-value is approximated by the proportion of the randomized F_{ST} values that are larger than the observed F_{ST} value. The F_{ST} statistic was first proposed in population genetics and could also be used to compare microbial communities. This statistic was used to compare the genetic diversity differences among the human intestinal microbial communities. The calculation of diversity is based on the alignment of these sequences, which makes this statistic unsuitable for comparing some community samples with highly divergent sequences that cannot be aligned.

An alternative approach to compare microbial communities is the phylogenetic (P) test [17]. For this method, the phylogenetic tree is firstly constructed using the sequences from all microbial communities, and then, the difference between each pair of communities is measured using parsimony score (the minimum number of branches that can be changed to form two separate subtrees with nonoverlapping branches for the two communities). The lower the parsimony value is, the greater the difference between the communities is. The significance of the observed parsimony score is evaluated by randomization under the null hypothesis that the sequences in different communities are randomly distributed across the phylogenetic tree. Two randomization methods can be used for generating the distribution of parsimony scores under the null hypothesis. One is to assume that the community identities of individual sequences remain fixed and the evolutionary relationships among these sequences are randomized. The other is to assume that the phylogenetic tree is fixed and the community identities of individual sequences are randomized.

The phylogenetic (P) test has been widely used in comparing microbial communities considering the evolutionary relationships of these sequences. It has been used in evaluating the relationships among human intestinal soil and marine viral microbial communities. The limitation of this approach is that it does not consider the branch length in the phylogenetic tree when estimating the parsimony score.

3.2 UniFrac, W-UniFrac, VAW-UniFrac, and Generalized UniFrac for Comparing Microbial Communities

Other widely used community comparison measures, namely UniFrac [12], weighted UniFrac (W-UniFrac) [14], generalized UniFrac [3], and variance adjusted weighted UniFrac (VAW-UniFrac) [2], are also based on phylogenetic trees. To implement these measures, the phylogenetic tree (rooted and known branch lengths) of the sequences in all communities is constructed, and each sequence is labeled according to the community from which it arises. The comparison of each pair of communities is performed based on the phylogenetic subtree by keeping the leaf nodes that are only from these two communities. The UniFrac measures the distance between two communities by the fraction of lengths of the tree branches that lead to descendants from each single community, but not from both communities. UniFrac can be calculated as

$$\text{UniFrac} = \frac{\sum_{i=1}^n b_i |I(p_i^A > 0) - I(p_i^B > 0)|}{\sum_{i=1}^n b_i}, \quad (14)$$

where n is the number of branches in the tree, b_i is the length of branch i , p_i^A and p_i^B are the OTU fractions in branch i for community A and B , respectively, and $I(\cdot)$ is the indicator function. The higher value of the statistic indicates that the two communities are evolutionarily far apart, and thus the difference between the two communities is high. If the two communities are identical, they do not have independent evolutionary processes, and thus the UniFrac value is zero. If the two communities are completely separated in the phylogenetic tree, that is, they follow two independent evolutionary processes, the UniFrac value is one.

From the definition of UniFrac, it can be seen that it only considers whether an OTU appears in a community, but not the abundance of the OTU. If the sets of OTUs contained in the two communities are identical, then, regardless of whether the abundance of each OTU is different or not between the communities, the UniFrac value is zero. In some cases, the researchers are interested in the changes in OTU abundances in the communities, such as studying the changes in the distribution of human intestinal microbial communities under antibiotic treatment. In such scenarios, UniFrac is not an appropriate measure for comparing the samples.

W-UniFrac, which is defined based on UniFrac, takes the abundance information into consideration and weights each branch length by the difference of the fractions of OTUs belonging to the branch for the two communities. W-UniFrac can be calculated as

$$\text{W-UniFrac} = \frac{\sum_{i=1}^n b_i \times |p_i^A - p_i^B|}{\sum_{i=1}^n b_i \times (p_i^A + p_i^B)}, \quad (15)$$

with the same notations as in the definition of UniFrac. The denominator of W-UniFrac is equal to or larger than the numerator and equality holds when either $p_i^A = 0$ or $p_i^B = 0$ for all i , which means that the two communities are completely separated. The numerators of both UniFrac and W-UniFrac can be written as

$$\sum_{i=1}^n b_i \times \omega_i. \quad (16)$$

In W-UniFrac, $\omega_i = |p_i^A - p_i^B|$, and in UniFrac, $\omega_i = |\mathcal{A}_i - \mathcal{B}_i|$, where $\mathcal{A}_i = 1$, if there are sequences from sample A in branch i , and $\mathcal{A}_i = 0$, otherwise. The definition of \mathcal{B}_i is similar to that of \mathcal{A}_i for sample B .

Many studies used UniFrac and W-UniFrac for analyzing the relationships among microbial communities, such as investigating the relationships among intestinal microbial communities from children, adult, and human from different countries, the difference between the intestinal microbial communities from patients with inflammatory bowel diseases (IBD) and healthy individuals, the variation of the microbial communities from different human body sites, and the mammalian gut microbial communities from carnivores, omnivores, and herbivores. UniFrac and W-UniFrac can be implemented using online application [13] or the local software, such as QIIME2 [1]. An improved version of UniFrac (Striped UniFrac) was also developed for comparing microbial communities when the number of microbial communities is large [18].

W-UniFrac is primarily influenced by abundance changes along branches with large proportions and is less sensitive to the abundance changes on the branches with small proportions. To attenuate the weight on branches with large proportions, Chen et al. [3] proposed a new measure, generalized UniFrac, that uses the relative difference $|p_i^A - p_i^B|/(p_i^A + p_i^B)$ in its formulation. Generalized UniFrac can be calculated as

$$G\text{-UniFrac} = \frac{\sum_{i=1}^n b_i (p_i^A + p_i^B)^\alpha \left| \frac{p_i^A - p_i^B}{p_i^A + p_i^B} \right|}{\sum_{i=1}^n b_i (p_i^A + p_i^B)^\alpha}, \quad (17)$$

where $\alpha \in [0, 1]$ controls the contribution from different branches. W-UniFrac is a special case of the generalized UniFrac when $\alpha = 1$. The generalized UniFrac measure is more powerful in detecting abundance changes in moderately abundant lineages than UniFrac and W-UniFrac [3] with appropriately chosen α .

3.3 VAW-UniFrac for Comparing Communities

The definition of W-UniFrac does not consider the variance of the weight $\omega_i = |p_i^A - p_i^B|$ for the i th branch length. The true relationship between the communities

may not be well characterized if the variance of ω_i in W-UniFrac is ignored. Hence, Chang et al. [2] proposed to adjust the weight ω_i using its standard deviation. First, a phylogenetic tree with all the $A_T + B_T$ sequences in the two communities as leaves is constructed, where A_T and B_T are the total numbers of reads in communities A and B, respectively. Then, the variance of ω_i for each branch is estimated based on the null hypothesis that the community identity of each leaf is randomly labeled across the two communities.

For the i th branch of the phylogenetic tree, let A_i and B_i be the numbers of sequences in the i th branch from communities A and B, respectively, $m_i = A_i + B_i$ be the total number of sequences for this branch, and $m = A_T + B_T$ be the total number of sequences belonging to the phylogenetic tree. Then, A_T sequences are randomly chosen from the total m sequences on the phylogenetic tree and are labeled as being from community A. For other sequences, they are labeled as being from community B. Then, the number of sequences in the i th branch that belong to community A, A_i , can be modeled with a hypergeometric distribution with parameters (m_i, m, A_T) . The probability distribution of A_i is

$$P(A_i = k) = \frac{\binom{m_i}{k} \binom{m-m_i}{A_T-k}}{\binom{m}{A_T}}, \quad k = \max(0, m_i + A_T - m), \dots, \min(m_i, A_T). \quad (18)$$

Therefore,

$$E(A_i) = \frac{m_i A_T}{m}, \quad Var(A_i) = \frac{m_i A_T (m - A_T)(m - m_i)}{m^2(m - 1)}. \quad (19)$$

Let

$$t_i = \frac{A_i}{A_T} - \frac{B_i}{B_T} = \frac{A_i}{A_T} - \frac{m_i - A_i}{B_T} = A_i \left(\frac{1}{A_T} - \frac{1}{B_T} \right) - \frac{m_i}{B_T}. \quad (20)$$

Under the null hypothesis that the A_T sequences are randomly chosen from the m sequences, the expectation and variance of t_i can be calculated as

$$E(t_i) = 0, \quad Var(t_i) = \frac{m_i(m - m_i)}{A_T B_T (m - 1)}. \quad (21)$$

From the above formula, the variance adjusted weight (VAW) for the length of i th branch of the tree is

$$\omega_i = \frac{|t_i|}{\sqrt{Var(t_i)}} \propto \frac{|\frac{A_i}{A_T} - \frac{B_i}{B_T}|}{\sqrt{m_i(m - m_i)}}. \quad (22)$$

VAW-UniFrac can be defined as

$$\text{VAW-UniFrac} = \frac{\sum_{i=1}^n b_i \frac{|\frac{A_i}{A_T} - \frac{B_i}{B_T}|}{\sqrt{m_i(m-m_i)}}}{\sum_{i=1}^n b_i \frac{|\frac{A_i}{A_T} + \frac{B_i}{B_T}|}{\sqrt{m_i(m-m_i)}}}. \quad (23)$$

Both simulated and real data were used to evaluate the performances of UniFrac, W-UniFrac, and VAW-UniFrac in revealing the relationships of microbial communities [2]. The real data included microbial communities from human skin, mouse gut, and tropical forests. The results show that VAW-UniFrac has better performance than UniFrac and W-UniFrac [2].

The OTU-based methods can compare different microbial communities if they share a large fraction of OTUs. On the other hand, the phylogenetic tree-based approaches are applicable when the OTU composition of different microbial communities is highly heterogeneous. The phylogenetic tree-based methods also consider the evolutionary relationships of the OTUs from different communities, while the OTU-based methods ignore such relationships. With the increasing sequencing depth, a large number of sequenced reads are produced for each sample. It is challenging to construct the phylogenetic tree of sequences from all samples before using the phylogenetic-based methods. In addition, sequencing errors and sequencing length may also affect the accuracy of phylogenetic tree, which in turn affect the performance of the phylogenetic-based methods.

4 Alignment-Free Methods for the Comparison of Microbial Communities

With the rapid development of high-throughput sequencing technologies, whole metagenome shotgun sequencing (WMGS) becomes a powerful approach to investigate complex microbial communities and the relationship between them. Metagenomic data provide more complete information than 16S rRNA gene profiling for the microbial communities. However, the beta diversity measures for metagenomic data from different microbial samples are significantly understudied. The general approach to analyze metagenomic data is based on alignment or *de novo* assembly. The alignment-based methods use alignment algorithms, such as the Smith-Waterman algorithm and BLAST, to first map the sequencing reads to known microbial genomes or pathways in existing nucleic acid databases and then compare the abundance of different microbial organisms or functional categories between each pair of samples. However, the known microbial genomes and genes in existing databases are limited such that the aligned sequencing reads represent only a small fraction in the samples and a large fraction of reads cannot be mapped to known genomes or genes. Therefore, alignment-based analysis methods do not make full use of the information from shotgun reads data resulting in significant loss of information. Based on the current literature, the fraction of unaligned reads in human gut metagenomic samples is about 40%, while, in ocean samples, the fraction

is up to 50%. *De novo* assembly based methods are generally time-consuming and computationally expensive. It is challenging to assemble the metagenomic reads for the microbial genomes that share similar regions and the short reads are not long enough to resolve the ambiguity. Alignment-free methods that do not rely on reference genomes or *de novo* assembly are promising alternative approaches for metagenomic sample comparison. Such methods include d_2^S and d_2^* [21, 22], *CVTree* [19], and others [28, 29]. The alignment-free methods can make use of all the information from the metagenomic samples and, thus, are powerful approaches to analyze metagenomic samples.

In shotgun sequencing, a read can come from the forward strand or the reverse strand. Therefore, we consider all the reads in a sample and their complements when we count the numbers of occurrences of k -mers (k -grams, words) in the sample. For a given k -mer w , let X_w be the number of its occurrences and $f_w^X = X_w / \sum_{w \in A^k} X_w$ be its relative frequency.

For some alignment-free dissimilarity measures, such as d_2^* and d_2^S [21, 22], and *CVTree* [19], the expected number of occurrences of word $w = w_1 w_2 \cdots w_k$, $E(X_w)$, can be estimated using an r th order Markov model [20]. The transition probability matrix for the Markov model can be estimated based on the numbers of occurrences of r -mers and $(r+1)$ -mers, and the estimated probability of observing nucleotide w_{r+1} given the preceding nucleotides $w_1 w_2 \cdots w_r$ is $P_M(w_{r+1} | w_1 w_2 \cdots w_r) = X_{w_1 w_2 \cdots w_{r+1}} / X_{w_1 w_2 \cdots w_r}$. Then, $E(X_w)$ based on the r th order Markov model can be calculated as

$$E(X_w) = N f_{w_1 w_2 \cdots w_r}^X \prod_{n=1}^{k-r} P_M(w_{n+r} | w_n w_{n+1} \cdots w_{n+r-1}), \quad (24)$$

where N is the total number of k -mers in a metagenomic sample. For the comparison of metagenomes, the independent identically distributed (IID) model ($r = 0$) with k -mer length between 6 and 9 bps works quite well [6]. The difference between the number of occurrences of k -mer w and its expectation is defined as $\tilde{X}_w = X_w - E(X_w)$.

The alignment-free dissimilarity measures, d_2 , d_2^* , and d_2^S , can be defined as

$$d_2 = \frac{1}{2} \left(1 - \frac{\sum_w X_w Y_w}{\sqrt{\sum_w X_w^2} \sqrt{\sum_w Y_w^2}} \right), \quad (25)$$

$$d_2^* = \frac{1}{2} \left(1 - \frac{\sum_w \frac{\tilde{X}_w}{\sqrt{E(\tilde{X}_w)}} \frac{\tilde{Y}_w}{\sqrt{E(\tilde{Y}_w)}}}{\sqrt{\sum_w \frac{\tilde{X}_w^2}{E(\tilde{X}_w)}} \sqrt{\sum_w \frac{\tilde{Y}_w^2}{E(\tilde{Y}_w)}}} \right), \quad (26)$$

$$d_2^S = \frac{1}{2} \left(1 - \frac{\sum_w \frac{\tilde{X}_w \tilde{Y}_w}{\sqrt{\tilde{X}_w^2 + \tilde{Y}_w^2}}}{\sqrt{\sum_w \frac{\tilde{X}_w^2}{\sqrt{\tilde{X}_w^2 + \tilde{Y}_w^2}}} \sqrt{\sum_w \frac{\tilde{Y}_w^2}{\sqrt{\tilde{X}_w^2 + \tilde{Y}_w^2}}}} \right), \quad (27)$$

where X and Y represent the vectors consisting of the numbers of occurrences of all the k -mers for the two metagenomic samples, respectively. The three dissimilarity measures were originally developed for studying the relationships among individual genomic sequences based on NGS short reads data [21]. They were later used to compare metagenomic samples [6, 23]. When the metagenomic samples are highly similar, the values of d_2 , d_2^* , and d_2^S are close to 0. The performances of these dissimilarity measures were evaluated using different types of metagenomic datasets, including mammalian gut metagenomic samples and marine metagenomic samples across the world [6]. The dissimilarity measure, d_2^S , can obtain superior performance than other measures when comparing metagenomic samples by revealing the group or environmental gradient relationships. In addition, the d_2^S was also used to analyze the 16S rRNA datasets from the gut microbiota of abalone for revealing the effect of temperature on their gut microbial composition [26]. The implementation of these alignment-free dissimilarity measures and many other alignment-free sequence comparison measures is available from the software, CAFE [16], with an excellent user interface.

The number of parameters in a Markov chain model increases exponentially with the order of the Markov chain r . With limited amount of data, the transition probabilities cannot be estimated accurately, and thus, the expected numbers of occurrences of k -mers cannot be estimated accurately either. To overcome such issues, Liao et al. [10] developed a data-driven variable length Markov chain (VLMC) method to estimate the expected numbers of occurrences of k -mers. The VLMC approach for estimating the expected numbers of occurrences of k -mers based on high-throughput metagenomic data is implemented with the following three steps: (1) a full prefix tree is built based on k -mers of different lengths; however, such a tree usually overfits the data. (2) The tree is subsequently pruned to remove redundant branches based on the Kullback–Leibler divergence. (3) Transition probabilities are calculated with respect to the Markov orders from the context tree. Finally, the expected numbers of occurrences of k -mers can be calculated. Compared with FOMC (fixed order Markov chain), the VLMC has a flexible number of parameters avoiding estimating the large number of free parameters. The modified d_2^* and d_2^S were applied to compare transcriptomic or metatranscriptomic datasets and have been shown to perform better than the corresponding measures based on the FOMC approach. However, when applied to metagenomic datasets, the modified d_2^* and d_2^S did not show a clear advantage over the original ones.

In addition to using VLMC for estimating the expected numbers of k -mers as above, Song et al. [23] proposed a reads binning approach to improve the performance of alignment-free dissimilarity measures in metagenomic sample comparison. The basic idea is to group the reads into different bins and then sum over the d_2^* and d_2^S across all the bins. For this approach, the first step is to group the bacterial genomic sequences into different bins and construct a corresponding Markov model for each bin using these bacterial genomic sequences. Secondly, the log-likelihood of each NGS read under a Markov chain of order r can be calculated as

$$LL(X_w | M_r) = \sum_{i=1}^{N-r} \log P_{M_r}(X_{w_{i+r}} | X_{w_i w_{i+1} \dots w_{i+r-1}}), \quad (28)$$

where N is the length of the read, M_r is an r th order Markov chain, and $P_{M_r}(X_{w_{i+r}} | X_{w_i w_{i+1} \dots w_{i+r-1}})$ can be estimated based on the numbers of occurrences of r -mers and $(r+1)$ -mers, as above. So, the bin which a read belongs to can be estimated as the one having the largest log-likelihood value for the read,

$$\lambda = \operatorname{argmax}_{c=1, \dots, C} LL(X | M_r^c), \quad (29)$$

where C is the number of Markov models constructed for reads binning, and λ is the predicted bin which the read belongs to. Finally, the k -mer count and its expectation are calculated in each bin of the NGS reads. The centralized k -mer counts from all the bins are combined and used for the extended definition of alignment-free dissimilarity measures,

$$\bar{X}_w = \sum_{c=1}^C (X_w^c - n_w^c p_{X,w}^c), \quad (30)$$

where c is the index of the c th bin.

Song et al. [23] evaluated the performances of d_2^* and d_2^S with Markov model-based binning approaches using both simulated and real metagenomic datasets. One hundred randomly chosen bacterial genomic sequences were used to simulate the NGS metagenomic data for two models, environmental gradient and group relationship. In the simulation study, it was shown that d_2^* and d_2^S with reads binning outperform the corresponding measures without reads binning or other binning approaches, such as COCACOLA [15], MetaBAT [7], Kraken [27], or MBMC [25], in detecting the relationship among metagenomic samples. The newly developed measures were used for analyzing three real metagenomic datasets, including 107 fecal metagenomic samples from different countries, 60 metagenomic samples from four human body sites, and 16 soil metagenomic samples from different ecosystems. The d_2^* and d_2^S with reads binning can successfully reveal the underlying relationships among these metagenomic samples: (a) the human gut metagenomic samples from different countries can be clustered according to country; (b) the samples from different human body sites can be clustered according to the body sites; and (c) metagenomic samples from different ecosystems can be clustered according to the environment.

However, the d_2^* and d_2^S dissimilarity measures with reads binning have several limitations. Firstly, their performances depend on the choice of the number of bins. Secondly, the optimal length of k -mers depends on the sequencing depth. Finally, the order of Markov models used to estimate the expected numbers of k -mers influences their performances.

5 A Tutorial on the Use of UniFrac Type and Alignment-Free Dissimilarity Measures for the Comparison of Metagenomic Samples

5.1 Analysis Steps for UniFrac, W-UniFrac, Generalized UniFrac, and VAW-UniFrac

In this section, we demonstrate the use of MEGA [9] to generate the tree linking the 16S rRNA sequences and the R package “GUniFrac” to calculate the various UniFrac measures including UniFrac, W-UniFrac, generalized UniFrac, and VAW-UniFrac. Fifteen samples with 16S rRNA sequences from three human body sites (oral, gastrointestinal, and skin) are used [5]. First, we use the option “Open A File” in MEGA to open the file containing the 16S rRNA sequences. Then, we use the option “Align” for multiple sequence alignment. The resulting multiple sequence alignment is stored in a file with “.meg” format. Finally, we use the option “Phylogeny” and select the method of “Neighbor Joining” to construct the phylogenetic tree for these 16S rRNA sequences. The resulting phylogenetic tree is stored in a file with “.tree” format.

With the rooted phylogenetic tree and the OTU count table for the 16S rRNA sequences from different samples, we use the R package “GUniFrac” to compute the UniFrac-type dissimilarity measures. The package can be installed in the R software using the command:

```
install.packages("GUniFrac")
```

The following steps are used for the analysis of the 16S rRNA sequence data.

1. Calculate the UniFrac type dissimilarity measures by

```
unifrac = GUniFrac(otu.tab, throat.tree, alpha=c(0, 0.5, 1))$unifrac  
# otu.tab is the OTU count table with n samples and q OTUs  
# throat.tree is the rooted phylogenetic tree generated by MEGA  
# alpha is the parameter controlling weight on the lineages
```

We can obtain the distance matrices of UniFrac-type measures using the following commands:

```

dw = unifrac[, , "d_1"] # Weighted UniFrac
du = unifrac[, , "d_UW"] # Unweighted UniFrac
dv = unifrac[, , "d_VAW"] # Variance adjusted weighted UniFrac
d0 = unifrac[, , "d_0"] # GUniFrac with alpha 0
d5 = unifrac[, , "d_0.5"] # GUniFrac with alpha 0.5

```

2. Visualization of the relationships among the samples using principal coordinates analysis (PCoA), which is a multidimensional scaling (MDS) method that converts a between-sample dissimilarity matrix into two-dimensional, or three-dimensional, ordination of samples and arranges the samples in the ordinate space. We used the R package “MASS” for PCoA:

```

library(MASS)
d = as.dist(du)
d.mds = isoMDS(d,k=2)
x1 = d.mds$points[,1]
x2 = d.mds$points[,2]
plot(x1,x2,pch=15)

```

3. The relationships among these samples can be visualized using scatterplots as shown in Fig. 1.

5.2 Analysis Steps for the Comparison of Microbial Communities Based on Alignment-Free Methods

CAFE [16] was developed to calculate 28 alignment-free genome and metagenome comparison measures and to visualize the relationships among the samples using either a clustering tree or PCoA. It can be easily installed on PC, Mac, and Unix machines. Here, we give an example of computing dissimilarity measures based on high-throughput sequencing data of eight microbiome samples from four body sites: buccal mucosa, supragingival plaque, tongue dorsum, and stool [11]. Since CAFE can only input files in FASTA format, we use a Python script for format conversion. The following commands are used for computing the values of d_2^* and d_2^S :

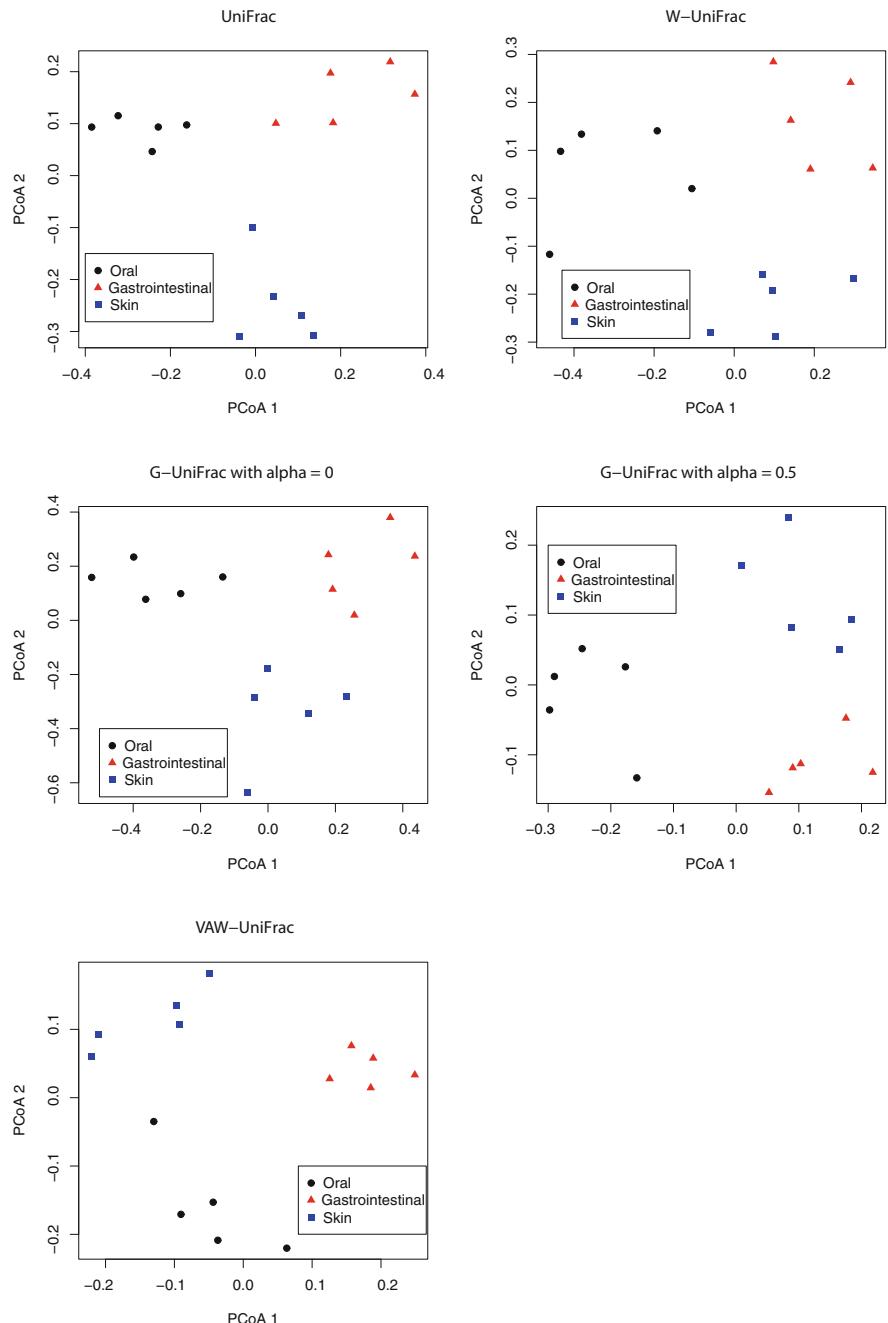


Fig. 1 The principal coordinate analysis (PCoA) plots of 15 samples from three body sites (oral, gastrointestinal, and skin) based on UniFrac, W-UniFrac, VAW-UniFrac, and G-UniFrac with $\alpha = 0$ and 0.5

```
./cafe -D D2shepp, D2star -I fa_files.fa -K 10 -M 3
# -D is the dissimilarity measures
# -I is the input of fasta files
# -K is the length of k-mer
# -M is the order of Markov chain
```

We then use the d_2^* and d_2^S values between each pair of samples to construct the cluster tree as shown in Fig. 2.

6 Discussion

In this chapter, we reviewed a variety of different beta diversity measures for comparing microbial communities using either 16S rRNA or NGS short reads data. Many measures have been developed for the comparison of microbial communities based on marker gene profiling data with UniFrac and Bray–Curtis dissimilarity being the most widely used ones in the literature. Many ecological insights about the relationships among microbial communities have been obtained through such analyses. For the comparison of microbial communities based on shotgun metagenomic short reads data, one common approach is to map the reads to known genomes, genes, and pathways to obtain their relative abundance profiles and then apply commonly used dissimilarity measures such as Bray–Curtis, Manhattan, and Euclidean

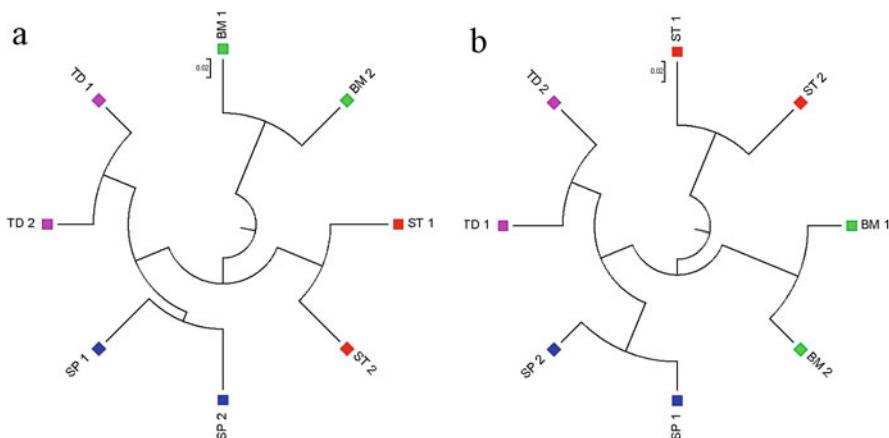


Fig. 2 The clustering results of the eight metagenomic samples using d_2^* (a) and d_2^S (b). Red squares: Stool (ST); Green squares: Buccal mucosa (BM); Blue squares: Supragingival plaque (SP); Purple squares: Tongue dorsum (TD)

distances to the abundance profiles to compare the metagenomic samples. Since a large fraction of short reads cannot be mapped to the known genomes or gene databases, these unmapped reads were not used in such methods for metagenome comparison resulting in potential loss of information. In the last decades, several research groups developed alignment-free methods using the frequency of k -mers for metagenome comparison including those based on the relative abundance profiles of k -mers or background adjusted k -mer frequencies. Applications of these statistics, in particular, the background adjusted frequency-based measures such as *CVTree*, d_2^* , and d_2^S , to both simulated and real metagenomes showed the power of these newly developed measures for metagenome comparison. Despite the advantages of alignment-free metagenome comparison methods, it is challenging to decide the optimal k -mer length and the background model for describing the microbial communities. It is also challenging to identify which microorganisms drive the difference of the microbial communities. These are the topics for future studies.

Acknowledgments This chapter was supported by the National Natural Science Foundation of China (11701546).

References

1. Bolyen, E., Rideout, J.R., Dillon, M.R., Bokulich, N.A., Abnet, C.C., Alghalith, G.A., Alexander, H., Alm, E.J., Arumugam, M., Asnicar, F.: Reproducible, interactive, scalable and extensible microbiome data science using QIIME 2. *Nature Biotechnology* **37**, 852–857 (2019)
2. Chang, Q., Luan, Y.H., Sun, F.Z.: Variance adjusted weighted UniFrac: a powerful beta diversity measure for comparing communities based on phylogeny. *BMC Bioinformatics* **12**(1), 118 (2011)
3. Chen, J., Bittinger, K., Charlson, E.S., Hoffmann, C., Lewis, J.D., Wu, G.D., Collman, R.G., Bushman, F.D., Li, H.Z.: Associating microbiome composition with environmental covariates using generalized UniFrac distances. *Bioinformatics* **28**(16), 2106–2113 (2012)
4. Excoffier, L., Lischer, H.E.L.: Arlequin suite ver 3.5: a new series of programs to perform population genetics analyses under Linux and Windows. *Mol. Ecol. Resour.* **10**, 564–567 (2010)
5. Huttenhower, C., Gevers, D., Knight, R., Abubucker, S., Badger, J.H., Chinwalla, A.T., Creasy, H.H., Earl, A.M., FitzGerald, M.G., Michael, G., Fulton, R.S., others: Structure, function and diversity of the healthy human microbiome. *Nature* **486**(7402), 207–214 (2012)
6. Jiang, B., Song, K., Ren, J., Deng, M.H., Sun, F.Z., Zhang, X.G.: Comparison of metagenomic samples using sequence signatures. *BMC Genomics* **13**, 730 (2012)
7. Kang, D.D., Froula, J., Egan, R., Wang, Z.: MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. *PeerJ* **3**, e1165 (2012)
8. Kuczynski, J., Liu, Z., Lozupone, C., McDonald, D., Fierer, N., Knight, R.: Microbial community resemblance methods differ in their ability to detect biologically relevant patterns. *Nature Methods* **7**(10), 813–819 (2010)
9. Kumar, S., Nei, M., Dudley, J., Tamura, K.: MEGA: a biologist-centric software for evolutionary analysis of DNA and protein sequences. *Brief. Bioinform.* **9**(4), 299–306 (2008)
10. Liao, W., Ren, J., Wang, K., Wang, S., Zeng, F., Wang, Y., Sun, F.Z.: Alignment-free transcriptomic and metatranscriptomic comparison using sequencing signatures with variable length Markov chains. *Scientific Reports* **6**, 37243 (2016)

11. Lloyd-Price, J., Mahurkar, A., Rahnavard, G., Crabtree, J., Orvis, J., Hall, A.B., Brady, A., Creasy, H.H., Mccracken, C., Giglio, M.G.: Strains, functions and dynamics in the expanded Human Microbiome Project. *Nature* **550**, 61–66 (2017)
12. Lozupone, C., Knight, R.: UniFrac: a new phylogenetic method for comparing microbial communities. *Appl. Environ. Microbiol.* **71**, 8228–8235 (2005)
13. Lozupone, C., Hamady, M., Knight, R.: UniFrac - An online tool for comparing microbial community diversity in a phylogenetic context. *BMC Bioinformatics* **7**, 371 (2006)
14. Lozupone, C.A., Hamady, M., Kelley, S.T., Knight, R.: Quantitative and qualitative β diversity measures lead to different insights into factors that structure microbial communities. *Appl. Environ. Microbiol.* **73**(5), 1576–1585 (2007)
15. Lu, Y.Y., Chen, T., Fuhrman, J.A., Sun, F.Z.: COCACOLA: binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment and paired-end read LinkAge. *Bioinformatics* **33**, 791–798 (2017a)
16. Lu, Y.Y., Tang, K.J., Ren, J., Fuhrman, J.A., Waterman, M.S., Sun, F.Z.: CAFE: aCcelerated Alignment-FrEe sequence analysis. *Nucleic Acids Res.* **45**, W554–W559 (2017b)
17. Martin, A.P.: Phylogenetic approaches for describing and comparing the diversity of microbial communities. *Appl. Environ. Microbiol.* **68**, 3673–3682 (2002)
18. McDonald, D., Vázquez-Baeza, Y., Koslicki, D., McClelland, J., Reeve, N., Xu, Z., Gonzalez, A., Knight, R.: Striped UniFrac: enabling microbiome analysis at unprecedented scale. *Nature Methods* **15**, 847–848 (2018)
19. Qi, J., Luo, H., Hao, B.L.: CVTree: a phylogenetic tree reconstruction tool based on whole genomes. *Nucleic Acids Res.* **32**, W45–W47 (2004)
20. Ren, J., Song, K., Deng, M.H., Reinert, G., Cannon, C.H., Sun, F.Z.: Inference of Markovian properties of molecular sequences from NGS data and applications to comparative genomics. *Bioinformatics* **32**(7), 993–1000 (2016)
21. Song, K., Ren, J., Zhai, Z.Y., Liu, X.M., Deng, M.H., Sun, F.Z.: Alignment-free sequence comparison based on next-generation sequencing reads. *J. Comput. Biol.* **20**, 64–79 (2013)
22. Song, K., Ren, J., Reinert, G., Deng, M.H., Waterman, M.S., Sun, F.Z.: New developments of alignment-free sequence comparison: measures, statistics and next-generation sequencing. *Brief. Bioinform.* **15**, 343–353 (2014)
23. Song, K., Ren, J., Sun, F.Z.: Reads binning improves alignment-free metagenome comparison. *Front. Genet.* **10**, 1156 (2019)
24. Vilella, A.J., Blanco-Garcia, A., Hutter, S., Rozas, J.: VariScan: Analysis of evolutionary patterns from large-scale DNA sequence polymorphism data. *Bioinformatics* **21**, 2791–2793 (2005)
25. Wang, Y., Hu, H., Li, X.: MBMC: An effective Markov chain approach for binning metagenomic reads from environmental shotgun sequencing projects. *Oomics J. Integr. Biol.* **20**, 470–479 (2016)
26. Wang, X., Tang, B., Luo, X., Ke, C., Huang, M., You, W., Wang, Y.: Effects of temperature, diet and genotype-induced variations on the gut microbiota of abalone. *Aquaculture* **524**, 735269 (2020)
27. Wood, D.E., Salzberg, S.L.: Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology* **15**, R46 (2014)
28. Zielezinski, A., Vinga, S., Almeida, J., Karlowski, W.M.: Alignment-free sequence comparison: benefits, applications, and tools. *Genome Biology* **18**, 186 (2017)
29. Zielezinski, A., Girgis, H.Z., Bernard, G., Leimeister, C.-A., Tang, K., Dencker, T., Lau, A.K., Rohling, S., Choi, J., Waterman, M.S., others: Benchmarking of alignment-free sequence comparison methods. *Genome Biology* **20**, 144 (2019)

Beta Diversity and Distance-Based Analysis of Microbiome Data



Anna M. Plantinga and Michael C. Wu

1 Introduction

Ecological measures of beta diversity aim to capture global dissimilarity between two ecological communities. In the context of microbiome data analysis, this corresponds to between-subject dissimilarities in microbial composition. Distance-based or “community-level” analysis then compares these pairwise dissimilarities between subjects to pairwise dissimilarities with respect to some phenotype. Scientifically, distance-based analysis identifies whether an association between microbiome composition and the phenotype is present; the answer to this question can justify further investigation into the form of that association. Scientists often view beta-diversity analysis as an investigation of differences in global microbial community structures rather than the role of individual community members.

Statistically, distance-based analysis has the potential for increased power, for at least three reasons. First, the association between any individual taxon and the phenotype may be of only modest strength, and looking at these associations in aggregate can enable detection of modest, but concerted, shifts. Second, depending on body site and taxonomic level of study, the microbiome may include hundreds to thousands of taxa, and power may be improved by avoiding the need to adjust for a large number of multiple comparisons. Third, taxa do not exist in isolation but rather have known phylogenetic relationships; other structural features, such as functional similarity, are under investigation and may be possible to incorporate

A. M. Plantinga
Williams College, Williamstown, MA, USA
e-mail: amp9@williams.edu

M. C. Wu (✉)
Fred Hutchinson Cancer Research Center, Seattle, WA, USA
e-mail: mcwu@fhcrc.org

in the future. Accounting for these relationships can allow the assumed form of association to better match the underlying truth and therefore again result in higher statistical power. Due to these scientific and statistical advantages, distance-based analysis of beta diversity has become one of the core features of most microbiome data analyses, along with within-sample diversity (alpha diversity) and investigation of the contribution of individual taxa.

Importantly, we note that beta-diversity analysis using distances and dissimilarities is not the only form of “community-level analysis.” For example, alpha diversity can also be viewed as a means of studying overall community compositions, and more recently, considerable work has been done on treating the microbial profiles as correlated high-dimensional response variables. Consequently, advanced multivariate modeling approaches such as those that can jointly model the distribution of all taxa have been proposed [32]. These methods are often based on generalized linear models, using random effects or latent variables to account for the high dimensionality and correlation between taxa [22, 23]. Joint modeling of abundance data can account for the mean–variance relationship and overdispersion in microbiome data and therefore may provide better separation of the locality and dispersion effects in ordination analysis; a fully specified model also permits assumptions underlying the analysis to be formally evaluated [32]. On the other hand, the trade-off is that they may make further assumptions and may be limited in terms of their scope of application, for example, focusing primarily on clustering or on hypothesis testing. However, although these methods represent an increasingly important topic, they remain outside the scope of this chapter, which focuses on the usual beta-diversity analysis approaches that remain the mainstay of many applications.

In the following sections, we outline common distance and dissimilarity measures (Sect. 2), unconstrained ordination analysis and graphical displays of beta diversity (Sect. 3), and approaches to hypothesis testing (Sect. 4). We conclude with a discussion of strengths, limitations, and directions for future investigation in distance-based microbiome analysis.

2 Quantifying Dissimilarity: Common Beta Diversity Metrics

Beta diversity, broadly speaking, refers to variation in species composition between different ecological (sub)communities. The idea of partitioning overall “landscape” species diversity, or gamma diversity, into the product of species complexity within particular niches—alpha diversity—and the extent of differentiation between niche communities—beta diversity—originated with R.H. Whittaker in 1960 [33, 34, 37]. In an ecological context, the “niche” often refers to a particular position along a resource gradient; the classical examples are varying levels of sunlight at different heights in a forest and nutrient density gradients in soil. In the context of human microbiome studies, a niche commonly refers to a particular body site on an individual, such as the intestinal or vaginal mucosa. Therefore, measures of beta

diversity in this context quantify dissimilarity between different body sites or individuals.

Before discussing specific measures of distance and dissimilarity, it is helpful to consider the precise meaning of these terms. To be a true distance, or a metric, a function d must satisfy the following three axioms:

1. Identity of indiscernibles (coincidence): $d(a, b) = 0 \iff a = b$;
2. Symmetry: $d(a, b) = d(b, a)$; and
3. Triangle inequality: $d(a, c) \leq d(a, b) + d(b, c)$.

While the first two properties hold for all indices we consider, the triangle inequality fails for several commonly used dissimilarities. Additional properties that may be useful in the evaluation of dissimilarities and distances are that the dissimilarity between two samples should not depend on scale changes, species that are absent from both samples, and the addition of new samples to an analysis (see [11] for further discussion).

Many approaches to quantifying dissimilarity between two microbial communities exist, of which several are summarized in Table 1. The Bray–Curtis dissimilarity is one of the most commonly used non-phylogenetic measures [6], though it does not satisfy the triangle inequality and hence is not a true distance. Typically applied to relative abundance data, the Bray–Curtis dissimilarity is the quantitative counterpart to the Sørensen–Dice coefficient, a “qualitative” dissimilarity in that it considers only presence or absence of species. The Sørensen–Dice coefficient is directly related to the Jaccard distance, another commonly used metric applicable to generic set comparisons, via $S = 2J/(1 + J)$, where S and J indicate the corresponding similarity indices $S = 1 - D^S$ and $J = 1 - D^J$. The Jaccard distance satisfies all three distance axioms. The canonical form of the Jaccard distance is qualitative, but a quantitative version is also used in practice.

Some distances and dissimilarities have also been developed specifically in the context of the microbiome, most notably the UniFrac family of distances and dissimilarities. These include unweighted UniFrac distance (qualitative) [19], weighted UniFrac (quantitative) [20], and generalized UniFrac (intermediate) [9], as well as a variance-adjusted weighted UniFrac [7]. The main advantage of the UniFrac family of dissimilarities is that phylogenetic information is directly incorporated, such that communities containing taxa with close phylogenetic relationships are considered more similar than those containing phylogenetically distant taxa.

Application of these distances and dissimilarities is affected by unique challenges presented by microbiome data. Qualitative measures often assume that sample “volume” is similar across the samples being compared, an assumption violated by uneven sampling depth. Since the total read count for each sample is not informative about the total bacterial concentration in a subject, rarefaction is often used prior to calculating qualitative dissimilarities in order to avoid confounding by sampling depth [15]. In addition, the compositionality of relative abundance data renders direct application of Euclidean distances concerning. Instead, Euclidean distances may be applied to centered log-ratio transformed abundances (the result is referred to as the Aitchison distance [2]), and centered log-ratio transformations

Table 1 Definitions and characteristics of commonly used distances and dissimilarities

Name	Definition	Data type	Phylogeny
Bray–Curtis	$D_{ii'}^{BC} = \frac{1}{2} \sum_{j=1}^J p_{ij} - p_{i'j} $	Quantitative ^a	No
Sørensen–Dice	$D_{ii'}^S = 1 - \frac{2 \sum_{j=1}^J \delta_{ij} \delta_{i'j}}{\sum_{j=1}^J \delta_{ij} + \sum_{j=1}^J \delta_{i'j}}$	Qualitative ^b	No
Binary Jaccard	$D_{ii'}^{JB} = 1 - \frac{\sum_{j=1}^J \delta_{ij} \delta_{i'j}}{\sum_{j=1}^J (\delta_{ij} + \delta_{i'j} - \delta_{ij} \delta_{i'j})}$	Qualitative	No
Quantitative Jaccard	$D_{ii'}^{JQ} = 1 - \frac{\sum_{j=1}^J \min(p_{ij}, p_{i'j})}{\sum_{j=1}^J \max(p_{ij}, p_{i'j})}$	Quantitative	No
Unweighted UniFrac	$D_{ii'}^U = \frac{\sum_{j=1}^J b_j \delta_{ij} - \delta_{i'j} }{\sum_{j=1}^J b_j}$	Qualitative	Yes ^c
Weighted UniFrac	$D_{ii'}^W = \frac{\sum_{j=1}^J b_j p_{ij} - p_{i'j} }{\sum_{j=1}^J b_j (p_{ij} + p_{i'j})}$	Quantitative	Yes
Generalized UniFrac	$D_{ii'}^{(\alpha)} = \frac{\sum_{j=1}^J b_j (p_{ij} + p_{i'j})^\alpha \left \frac{p_{ij} - p_{i'j}}{p_{ij} + p_{i'j}} \right }{\sum_{j=1}^J b_j (p_{ij} + p_{i'j})^\alpha}$	Intermediate ^d	Yes

^a p_{ij} indicates the relative abundance of taxon j in subject i

^b $\delta_{ij} = I(p_{ij} > 0)$ indicates presence of taxon j in subject i

^c b_j indicates the length of the branch leading to taxon j on a rooted phylogenetic tree

^d The parameter α controls how much weight is given to quantitative versus qualitative information in the generalized UniFrac dissimilarity

are sometimes recommended prior to quantitative dissimilarities more generally, though it is unclear whether this is necessary for valid distance-based inference [31]. An isometric log-ratio (ILR) transformed variant of weighted UniFrac has been developed to account for compositionality in UniFrac family distances [28].

As suggested in the preceding paragraphs, two important factors to consider when choosing a dissimilarity for a particular analysis are (1) whether to use a quantitative (abundance) measure or a qualitative (presence/absence) measure and (2) whether to incorporate structural relationships between components of the community (primarily phylogeny). The goal is to choose a dissimilarity that closely matches the true form of association, to capture as much of the relevant structure as possible. A “good” choice of dissimilarity will lead to better discrimination on graphical displays and higher power in global hypothesis tests. Naturally, the underlying form of association is rarely known in advance. Therefore, beta-diversity analyses are often repeated with several different measures of dissimilarity, and most distance-based hypothesis testing frameworks incorporate omnibus tests that consider multiple dissimilarity measures in addition to tests that require choosing a single dissimilarity.

We now compute the distances and dissimilarities described in Table 1 using R. For all examples in the present chapter, we use the throat microbiome dataset of Charlson et al. [8], which is readily available in the R package GUniFrac. The OTU table includes abundance of 856 taxa in 60 individuals, of whom 32 are nonsmokers and 28 are smokers. Additional covariates such as age and sex and a phylogenetic tree relating the OTUs are also available.

Computing Distances and Dissimilarities

```

library(GUniFrac) # for UniFrac family distances
library(vegan)     # for all other distances

## Load Charlson data (in GUniFrac)
data(throat.otu.tab)
data(throat.tree)
data(throat.meta)

## Compute D matrices
# Creates array containing all requested alphas +
unweighted
unifrac <- GUniFrac(otu.tab = throat.otu.tab,
                     tree = throat.tree,
                     alpha = c(0.5, 1))$unifrac
unweighted <- unifrac[, "d_UW"]
gen0.5 <- unifrac[, "d_0.5"]
weighted <- unifrac[, "d_1"]

# Sorensen--Dice is binary Bray--Curtis
braycurtis <- as.matrix(vegdist(x = throat.otu.tab,
                                  method = "bray", binary = FALSE))
sorense <- as.matrix(vegdist(x = throat.otu.tab,
                             method = "bray", binary = TRUE))

# Quantitative and binary Jaccard
jaccard.q <- as.matrix(vegdist(x = throat.otu.tab,
                               method = "jaccard", binary = FALSE))
jaccard.b <- as.matrix(vegdist(x = throat.otu.tab,
                               method = "jaccard", binary = TRUE))

```

A sample of the output for the first four subjects is shown below and demonstrates that based on the weighted UniFrac measure of dissimilarity, the microbiomes of subjects 1 and 3 differ more from each other ($D_{1,3} = 0.30$) than the microbiomes of subjects 4 and 5 ($D_{4,5} = 0.13$). In fact, subject 1 is quite different from each of 3, 4, and 5, whereas those three subjects have more similar microbiomes. The metadata reveals that subject 1 is a nonsmoker, whereas subjects 3, 4, and 5 are all smokers. As we move on to ordination and hypothesis testing, the goal will be to more rigorously evaluate whether (as we saw in this small example) smokers and nonsmokers tend to be more dissimilar than subjects in the same group.

Weighted UniFrac Distances

```
> weighted[1:4,1:4]
      Subj1     Subj3     Subj4     Subj5
Subj1 0.0000000 0.3038448 0.2708932 0.2875414
Subj3 0.3038448 0.0000000 0.1431624 0.1823744
Subj4 0.2708932 0.1431624 0.0000000 0.1282493
Subj5 0.2875414 0.1823744 0.1282493 0.0000000
```

3 Ordination and Dimension Reduction

Ordination analysis aims to summarize as much of the variability in the original data as possible in a lower dimensional space [37]. Unconstrained ordination, in which key explanatory or response variables are not taken into account during the ordination analysis, is primarily an exploratory method; additional variables are included only in *post hoc* analyses. In contrast, constrained ordination computes the new axes conditional on the outcome or explanatory variables of interest with the aim of discovering microbiome features that best distinguish different levels of the constraining variable. In this section, we discuss several of the most common unconstrained ordination and dimension reduction methods, including subsequent graphical displays, for exploratory analysis of microbiome beta diversity.

3.1 Principal Coordinates Analysis

Principal coordinates analysis (PCoA), also called metric multidimensional scaling (MDS), is a standard ordination method for generic distance or dissimilarity measures. The standard PCoA analysis proceeds as follows:

1. Compute an $n \times n$ distance or a dissimilarity matrix \mathbf{D} . This may be generated using one of the distances or dissimilarities defined in Table 1 or any others deemed suitable for a particular application.
2. Center \mathbf{D} via

$$\mathbf{K} = -\frac{1}{2} \left(\mathbf{I} - \frac{\mathbf{1}_n \mathbf{1}'_n}{n} \right) \mathbf{D}^2 \left(\mathbf{I} - \frac{\mathbf{1}_n \mathbf{1}'_n}{n} \right), \quad (1)$$

where \mathbf{D}^2 is the elementwise square.

3. Compute the eigendecomposition of \mathbf{K} .

The eigenvectors corresponding to the largest m eigenvalues in the eigendecomposition of \mathbf{K} may then be used for a lower dimensional representation of the data or for graphical displays. In particular, the two leading eigenvectors (principal components or PCs) are often used to plot the data.

Each distance and dissimilarity emphasizes a different aspect of the data and community structure. Thus, the choice of distance or dissimilarity in Step 1 affects the features of the data displayed in the PCoA plot. For instance, if the presence of rare taxa varies between subjects with and without a disease, then a PCoA plot based on a qualitative measure such as unweighted UniFrac (which focuses on presence/absence) may show distinct separation between the two groups, whereas a PCoA plot based on a quantitative measure such as weighted UniFrac (which emphasizes relative abundance) would show little or no separation. Hence, as mentioned in the previous section, presenting the results under multiple dissimilarity metrics may be more robust to the true form of association and may even provide some information about which microbiome features are driving the association.

Below, we apply PCoA to the Charlson data. The analysis below demonstrates that the first two PCs in an ordinary PCoA analysis using the weighted UniFrac distance explain 29.4% and 22.5% of the variability, respectively. The PCoA plot is displayed in Fig. 1A, showing moderate visual separation between smokers and nonsmokers.

PCoA

```
library(ape)
library(ggplot2)

# PCoA analysis
pcres <- pcoa(weighted)

# Relative eigenvalues: percent variance explained
pctvar <- round(100*pcres$values$Relative_eig, 1)

# Ordinary PCoA plot
plotdat = data.frame(Axis1 = pcres$vectors[,1],
                      Axis2 = pcres$vectors[,2],
                      Smoke = throat.meta$SmokingStatus)
ggplot(data = plotdat,
       aes(x = Axis1, y = Axis2,
           color = Smoke, shape = Smoke)) +
  geom_point() +
  xlab(paste("Axis 1 (", pctvar[1], "%)", sep = "")) +
  ylab(paste("Axis 2 (", pctvar[2], "%)", sep = ""))
```

Confounding covariates can make associations of interest difficult to display on a PCoA plot. To adjust for mean shifts due to confounders, the centering matrix $\mathbf{I} - (\mathbf{1}_n \mathbf{1}'_n)/n$ in Eq. 1 may be replaced with a covariate adjusted centering matrix $(\mathbf{I} - \mathbf{H})$, where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the projection matrix in linear regression [27]. In the case of Euclidean distances, this simplifies to using the residuals from a linear model to construct the distance matrix.

Adjusted PCoA is available through the R package aPCoA or the corresponding R Shiny app. In the sample code below, we adjust the previous PCoA analysis for age and sex. The first two PCs in the adjusted PCoA analysis explain 28.0% and 23.3% of the variability, compared to 29.4% and 22.5% in the ordinary PCoA analysis. The similarity between these values is consistent with graphical results; Fig. 1b, the adjusted PCoA plot, does not display substantially clearer visual separation between groups than Fig. 1a.

aPCoA

```
library(aPCoA)
throat.meta$Female = as.numeric(throat.meta$Sex ==
                                  "Female")
apcres <- aPCoA(weighted ~ Age + Female,
                  data = throat.meta, maincov = SmokingStatus,
                  drawEllipse = FALSE, drawCenter = FALSE)
```

3.2 Double Principal Coordinate Analysis

Though similarly named, double principal coordinate analysis (DPCoA) takes a different theoretical approach [24]. Whereas PCoA plots are based on dissimilarities such as UniFrac distances between pairs of subjects, DPCoA defines the distance between two microbial communities as a combination of the within-community diversity (between species) and the between-community diversity. Mathematically, the distance between community i and community i' is $D_{ii'} = H_{ii'} - (H_i + H_{i'})/2$, where H_i is the average patristic distance between bacterial taxa in the same community and $H_{ii'}$ is the average distance between members of community i and community i' . Functionally, DPCoA tends to behave similarly to PCoA based on weighted UniFrac and can be generalized similarly to place more emphasis on rare taxa or less emphasis on phylogenetic structure [12].

DPCoA is available in the ade4 R package or in the phyloseq package if the data are stored in a phyloseq object. Output includes coordinates for both samples and taxa, so either or both may be displayed on ordination plots. For comparison with PCoA, we focus on ordination of samples in Fig. 1c. The first two PCs from a DPCoA analysis of the Charlson data explain 31.2% and 18.8% of variation.

DPCoA

```

library(ade4)
dpcoa.df = as.data.frame(throat.otu.tab)
patristicDist <- sqrt(as.dist(cophenetic.phylo(throat.
    tree)))
dpcores = dpcoa(df = dpcoa.df, dis = patristicDist,
                 scannf = FALSE, nf = 2)
pctvar.dpc <- 100*dpcores$eig/sum(dpcores$eig)

dpcores$dls    # Coordinates of taxa, first nf axes
dpcores$li     # Coordinates of samples

```

3.3 Biplots

As we will see again in the hypothesis testing section, a persistent difficulty with analyses based on distances and dissimilarities is identifying taxa that are key players in the global dissimilarity. Biplots display samples as points on a scatterplot but additionally plot the columns of a data matrix (here, bacterial taxa) as arrows on the graph [13]. This allows the separation between sample groups to be attributed to particular taxa and can therefore improve the interpretability of ordination plots.

Functionality for PCoA biplots is available via `biplot.pcoa()` in package `ape` or `envfit()` in package `vegan`. The biplot for PCoA with weighted UniFrac is displayed in Fig. 1d, displaying only the features that were most significant by `envfit()`'s permutation analysis. The figure shows that there are sets of OTUs that separate most smokers from most nonsmokers, and an additional group of OTUs that distinguish the outlying subjects. For example, OTUs 2572, 4703, 1490, 2434, and 3538 are more common in the cluster of smokers; OTUs 3954, 1549, 444, 3227, and 4871 are more common to nonsmokers. In this example, the interpretability of the ordination results is not greatly improved because no map is available to translate the OTU identifiers in the Charlson data into taxonomic identifiers, but in general these taxa could be further investigated to provide more specific information about the differences in the microbiome between groups.

PCoA Biplot

```

## Automated approach
library(ape)
biplot.pcoa(dpcores, Y = throat.otu.tab,
            rn = c("N", "S") [throat.meta$SmokingStatus] )

## Alternative approach, allows more control over PCoA

```

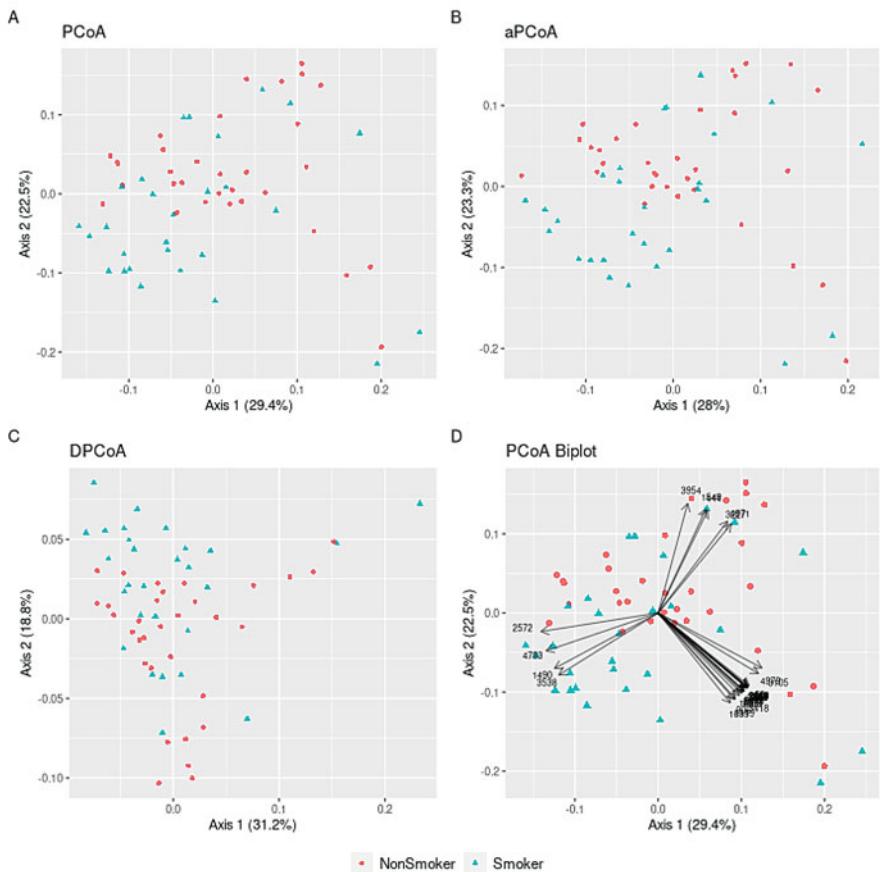


Fig. 1 Demonstration of several types of ordination plots. **(a)** Ordinary PCoA with weighted UniFrac distances. **(b)** aPCoA plot, adjusting for sex and age. **(c)** DPCoA ordination plot of samples. **(d)** PCoA biplot using weighted UniFrac distances, displaying arrows for features with permutation p -values < 0.0001

```

plot
# cmdscale() gives same PCoA results as pcoa()
library(vegan)
pcres2 <- cmdscale(weighted)
efit <- envfit(ord = pcres2, env = throat.otu.tab)
plot(pcres2[,2] ~ pcres2[,1],
     xlab = "Axis 1", ylab = "Axis 2",
     col = throat.meta$SmokingStatus)
plot(efit) # adds arrows to previous plot

```

3.4 Accounting for Compositionality

The compositional nature of microbiome data affects the patterns visible on ordination plots [14]. In particular, ordination based on quantitative distances and dissimilarities may place too much weight on the most abundant taxa rather than on those that best discriminate between samples, and those based on qualitative or binary distances and dissimilarities depend strongly on which features are included in the analysis (for example, preprocessing related to rare taxa or the sampling depth chosen for rarefaction) [35]. Compositionality exacerbates this problem due to negative correlation between features, which is further exacerbated by subsetting and aggregation of features during data preprocessing. The use of the PhILR transformation, which combines the ILR transformation for compositionality with phylogenetic information [28], or the Aitchison distance, which is the Euclidean distance applied to CLR-transformed data [2, 14], may better represent the structures in the underlying count data for clustering and ordination.

The graphical display corresponding to the Aitchison distance is the variance-based compositional principal component biplot [1], which tends to be less sensitive to the effect of very rare features and therefore more stable when using data subsets [35]. Similarly, PCA may be used to display Euclidean distances based on the PhILR transformation, and the PhILR-based distances may provide better clustering and classification behavior compared to phylogenetic approaches that are not composition-aware [28]. Notably, due to the ILR transformation, features identified on a biplot are transformed ratios of abundances (often called “balances”) rather than abundances of individual OTUs.

Most compositional transformations, including ILR and CLR, involve log transformations of relative abundances and therefore require first addressing zero counts. Zero replacement methods include adding a small pseudocount, which does not distinguish between true and sampling zeros, and more sophisticated multiplicative approaches such as Bayesian-multiplicative (BM) replacement with a variety of priors [21], used below.

Ordination and Plotting: Compositional Approaches

```
# Load packages
library(phiLR) # available on Bioconductor
library(compositions) # for CLR
library(zCompositions) # for cmultRepl

# Data preparation
# Bayesian multiplicative replacement (GBM) for zero replacement
# GBM requires observation in 2+ subj: exclude singlettons
sampcount <- apply(throat.otu.tab, 2, FUN = function(x) sum(x != 0))
twoplus <- which(sampcount > 1)
throat.gt1 <- throat.otu.tab[, twoplus]
throat.gbm <- cmultRepl(X = throat.gt1, label = 0, method = "GBM")
```

```
# Keep corresponding tree tips
tree.twoplus <- keep.tip(throat.tree, colnames(throat.gbm))

# PhILR transformation and corresponding distance
throat.philr <- philr(as.matrix(throat.gbm), tree = tree.twoplus)
D.philr <- as.matrix(dist(throat.philr, "euclidean"))

# Aitchison distance: CLR transformation -> Euclidean distance
throat.clr <- clr(x = throat.gbm)
D.aitch <- as.matrix(dist(throat.clr, "euclidean"))

# PCoA with Euclidean distance = PCA
pcres.philr <- pcoa(D = D.philr)
pcres.aitch <- pcoa(D = D.aitch)
```

Figure 2 displays the PCA biplots corresponding to PhILR-based distances (panel A) and the Aitchison distance (panel B). The PCA biplot using the Aitchison distance is largely similar to the PCoA biplot results using weighted UniFrac (Fig. 1a), producing one clearly separated cluster and two mildly distinct clusters. Each of the groups of taxa includes all or all except one of the taxa in the corresponding groups on the PCoA biplot of Fig. 1d, though many more microbiome features were highly significant using the compositional approaches (the p -value threshold for arrows included in Fig. 2 was 0.00001, compared to 0.0001 for Fig. 1d). The PhILR ordination differs most from the others, likely due to the selection of a new basis in the ILR transformation.

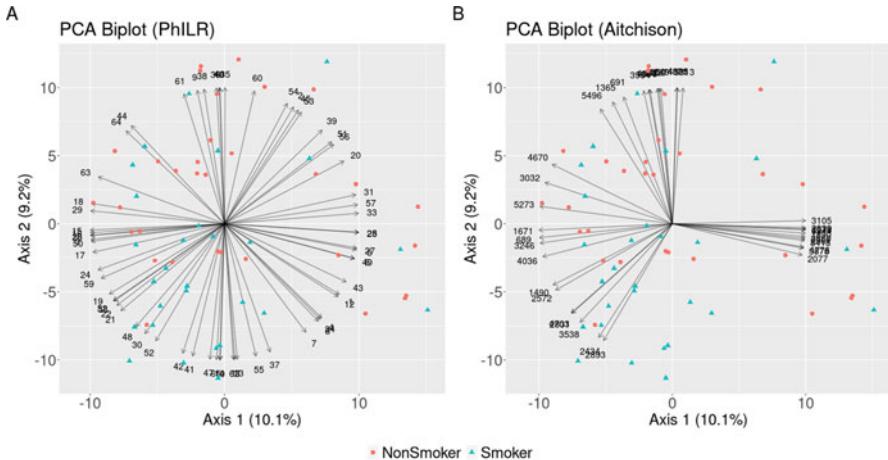


Fig. 2 Demonstration of ordination biplots that account for compositionality. Arrows are shown for features with permutation p -values < 0.00001 . **(a)** Euclidean distances applied following PhILR transformation. **(b)** Aitchison distance (Euclidean distances applied following CLR transformation)

3.5 Model-Based Ordination Using Latent Variables

The preceding ordination approaches are distance-based rather than model-based in the sense that they require choice of a beta diversity measure, but not of a statistical model for microbial abundances. An alternative model-based approach models the abundance of each microbial taxon as a function of unmeasured latent variables (for the purpose of ordination plots, usually two) using a generalized linear model framework [29, 38]. Careful choice of model and distributional assumptions allows this approach to account for dispersion effects, which are omnipresent due to the zero-inflated nature of microbiome data and may be conflated with mean shifts between groups in traditional ordination analyses. Similarly to PCA biplots, the estimated values of the latent variables provide subject-level ordination, whereas the factor loadings provide information about the contributions of individual taxa [32].

4 Distance-Based Hypothesis Testing

Visualization is useful for finding large systematic effects, but the top principal components often only explain a small portion of variability (second PC sometimes < 5%), so a lack of discrimination visually does not mean that there are no differences between groups. Also, because PCoA and similar approaches are unsupervised, it is not clear that the identified PCs are capturing any variability related to the phenotypes of interest. To more fully explore the association with a particular phenotype, therefore, we need to use all of the data, not just the top few axes of variation. We now turn our attention to global tests of association that do exactly this.

In the context of distance-based analysis, the null hypothesis under investigation is that there is no association between [dis]similarity in microbiomes and [dis]similarity in outcomes; that is, subjects with more similar microbiomes do not tend to have more similar phenotypes. In the following sections, we outline several approaches to formally testing variations on this hypothesis.

Some of these approaches were initially developed in the context of testing associations for other types of omics data, such as genetic variants in GWAS. The structure of the tests carried over well due to the shared characteristics of high-dimensional data, rare features, and modest effect sizes. However, to be appropriate for microbiome data, the tests needed to additionally accommodate extrinsic structure (here encoded as more complex distances and dissimilarities, some of which incorporate phylogenetic relationships among taxa) and generally smaller sample sizes..

Notationally, suppose throughout the following sections that n samples have been collected with information on the microbiome ($\mathbf{Z}_{n \times J}$), covariates or potential confounders ($\mathbf{X}_{n \times q}$), and the outcome variable \mathbf{y} . \mathbf{D} indicates an $n \times n$ matrix of

pairwise dissimilarities and may be constructed using one of the common measures described in Table 1 or an alternative. For the sake of simplicity (and because all of the following tests allow continuous phenotypes), we will suppose that y is continuous; the availability of each test for other outcome types is described in the sections below.

As mentioned in Sect. 2, the optimal choice of distance or dissimilarity \mathbf{D} for distance-based analyses depends on the true form of association between the microbiome and the outcome variable, including whether taxon presence or taxon abundance matters most and whether phylogenetic relationships among taxa should be considered. Distances that consider compositionality may be used as well. While compositionality of the microbiome may play a role in the power of an analysis—the impact of compositionality on the power of distance-based tests has not been rigorously evaluated as of this writing—it does not affect the type I error of distance-based tests. We therefore proceed to formal distance-based hypothesis testing, with the understanding that compositionally aware distances may be substituted for any distance matrices considered below as desired.

We begin by generating data that will be used throughout this section. To display the behavior of different kernels under different true forms of association, two outcome scenarios are considered: y_{clust} is associated with a moderately common phylogenetic cluster of OTUs (11.6% of all reads; includes 63 taxa, of which 53 have average abundance < 0.1%), and y_{common} is associated with three of the ten most common OTUs (11.3% of all reads). Based on the data generation strategy, a member of the UniFrac family of distances would be expected to perform best for y_{clust} , since phylogeny is informative about the associated OTUs. Bray–Curtis or quantitative Jaccard would be expected to provide higher power for y_{common} , since phylogeny does not matter but taxon abundance does.

Data Preparation

```
set.seed(1)
library(MiRKAT) # for D2K
Ds <- list(uw = unweighted, d5 = gen0.5, w = weighted,
           bc = braycurtis, s = sorensen)
Ks <- lapply(Ds, D2K)

# Covariates
X <- cbind(throat.meta$Age, throat.meta$Female)

# Prepare phylogenetic clustering
library(dirmult) # dirmult
library(cluster) # pam
dd = dirmult(throat.otu.tab)
nClus = 20 # clusters on phylogenetic tree
tree.dist = cophenetic(throat.tree)
```

```

obj <- pam(tree.dist, nClus) # partition around medoids
clust <- obj$clustering # cluster labels

# Associated OTUs: Phylogenetic cluster 9
assoc.clust <- throat.otu.tab[,names(which(clust == 9))]
total.clust <- scale(as.numeric(rowSums(assoc.clust)))

# Associated OTUs: 3 of the 10 most common taxa
tax.totals = apply(throat.otu.tab, 2, sum)
common10 = names(sort(tax.totals, decreasing = TRUE))
keep3 = common10[sample(1:10, 3)]
assoc.common <- throat.otu.tab[, keep3]
total.common <- scale(as.numeric(rowSums(assoc.common)))

# Generate outcome data
n.subj <- nrow(throat.meta)
b <- 1.0 # Moderately large effect size
y.base <- 0.5*(X[,1] + X[,2]) + rnorm(n.subj)
y.clust <- y.base + b*total.clust
y.common <- y.base + b*total.common

```

4.1 Permutation Tests

The first major approach to distance-based hypothesis testing is permutation-based testing. Permutational multivariate analysis of variance (PERMANOVA) is a semiparametric method that may be based on any chosen dissimilarity and tests the null hypothesis that the centroids of each group (in the space of the chosen dissimilarity) are equivalent [3, 37]. Formally, a pseudo-F statistic is calculated via

$$F = \frac{tr(\mathbf{HK})/(q-1)}{tr\{(\mathbf{I} - \mathbf{H})\mathbf{K}\}/(n-q)}, \quad (2)$$

where \mathbf{I} is the $n \times n$ identity matrix \mathbf{H} is the projection (hat) matrix, $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ for design matrix $\mathbf{X}_{n \times q}$ and \mathbf{K} is the centered dissimilarity matrix defined in Eq. 1. P -values are calculated by permutation. If Euclidean distances are used to define \mathbf{D} , then this is equivalent to a classical multivariate ANOVA with permutation P -values. As for all of the distance-based tests, the power for PERMANOVA is highest when the distance used captures the most information about the true association, for instance, focusing appropriately on rare or common taxa or incorporating phylogenetic information.

To account for confounders in PERMANOVA, \mathbf{X} may be partitioned by columns into m groups, $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_m)$, where each \mathbf{X}_k indicates a variable or a set of variables that should be tested jointly (e.g., indicators for a categorical variable). The Gram–Schmidt process is used to orthonormalize \mathbf{X} , which implies that P -values may differ depending on the order in which variables are added to the model, since each subsequent \mathbf{X}_k is transformed to be orthonormal to $(\mathbf{X}_1, \dots, \mathbf{X}_{k-1})$. To address this difficulty and permit ensemble testing with multiple distances or dissimilarities, PERMANOVA-S [30] regresses the predictor of interest \mathbf{X} on the potential confounders to generate residuals and uses these residuals in all subsequent testing. The PERMANOVA-S ensemble test again uses permutation to assess the significance of the minimum P -value across multiple dissimilarity measures.

PERMANOVA assumes exchangeability under the null hypothesis and homogeneity of dispersion across groups, though the test is robust to violations of the latter assumption in balanced designs and recent extensions permit heterogeneous dispersion across groups [5]. Because the power of the test is highly sensitive to the choice of distance \mathbf{D} , careful choice based on prior scientific knowledge or use of the PERMANOVA-S ensemble test is strongly recommended. In particular, power for PERMANOVA (and, indeed, all distance-based tests) is highest when the distance chosen best matches the true form of association.

PERMANOVA-S is implemented in C, but not in R, so the analysis below is restricted to PERMANOVA with single dissimilarities. The variable of interest is generally added to the model last, as shown below. The results displayed in Table 2 demonstrate the highest significance for $\mathbf{y}_{\text{clust}}$ with unweighted UniFrac, though the R -squared value is nearly identical between unweighted UniFrac and Bray–Curtis. For $\mathbf{y}_{\text{common}}$, Bray–Curtis is the most statistically significant one, due to the irrelevance of phylogenetic information and the importance of a few common taxa.

PERMANOVA

```
library(vegan)
perm.data <- data.frame(y.clust = y.clust,
                        y.common = y.common,
                        age = throat.meta$Age,
                        fem = throat.meta$Female)

# adonis and adonis2 perform PERMANOVA (single D)
# PERMANOVA-S is implemented in C but not R

adonis2(weighted ~ age + fem + y.clust, data = perm.data)
```

The analysis of similarities (ANOSIM) is an alternative permutation test based on ranked dissimilarities [10], testing the null hypothesis that

Table 2 PERMANOVA P -values using unweighted UniFrac, weighted UniFrac, and Bray–Curtis dissimilarities

Outcome	Kernel	P -value	R -squared
$\mathbf{y}_{\text{clust}}$	Unweighted UniFrac	0.001	0.037
	Weighted UniFrac	0.057	0.033
	Bray–Curtis	0.012	0.038
$\mathbf{y}_{\text{common}}$	Unweighted UniFrac	0.211	0.020
	Weighted UniFrac	0.020	0.039
	Bray–Curtis	0.001	0.043

the average of the *ranks* of within-group distances is greater than or equal to the average of the *ranks* of between-group distances;

however, ANOSIM is highly sensitive to differences in dispersion across groups and generally has lower power than PERMANOVA [4].

4.2 Kernel Machine Regression Tests

The second family of distance-based tests for microbiome data is the set of microbiome regression-based kernel association tests (MiRKATs) [43], which are based on approaches used in genetic association studies. Like PERMANOVA-S, MiRKAT allows adjustment for confounders and choice of one or several distance and dissimilarity measures. However, MiRKAT is computationally more efficient due to the use of a variance component score test. In addition, for complex study designs, though appropriate permutation strategies often exist, they are less easily attainable for many applied scientists than the use of regression-based tests. MiRKAT family tests allow binary, censored time-to-event, multivariate, structured high-dimensional, and other data types for \mathbf{y} [25, 41, 42]. However, for ease of exposition, we assume a quantitative (continuous) \mathbf{y} with the understanding that the approach generalizes to more sophisticated outcomes.

For a quantitative outcome, MiRKAT uses a linear kernel machine regression model,

$$y_i = \beta_0 + \boldsymbol{\beta}' \mathbf{X}_i + f(\mathbf{Z}_i) + \epsilon_i, \quad (3)$$

where $f(\cdot)$ is a function that fully describes the relationship between the microbiome and the outcome and ϵ_i is an error term with mean 0 and variance σ^2 . To test the association between the microbiome and the outcome, then, is to test $H_0 : f(\mathbf{Z}) = 0$. We assume $f(\mathbf{Z}_i) \in \mathcal{H}_k$, a reproducing kernel Hilbert space generated from a kernel function $K(\cdot, \cdot)$, such that $f(\mathbf{Z}_i) = \sum_{i'=1}^n \alpha_{i'} K(\mathbf{Z}_i, \mathbf{Z}_{i'})$.

The kernel function is a measure of pairwise similarity between individuals and is typically constructed by transforming a measure of distance or dissimilarity (refer

to Table 1 for several common examples) into a similarity matrix using Eq. 1. Using and Euclidean distance corresponds to a linear kernel $K(\mathbf{Z}_i, \mathbf{Z}_{i'}) = \sum_{j=1}^J Z_{ij} Z_{i'j}$ and therefore assumes a linear form of association between the microbiome and the outcome, $f(\mathbf{Z}_i) = \sum_{j=1}^J Z_{ij} \gamma_j$. Using more complex kernels allows more complex forms of association.

A key relationship between kernel machine regression and linear mixed models [18] is that the model in (3) is equivalent to the mixed model:

$$\mathbf{y} = \beta_0 + \mathbf{X}\boldsymbol{\beta} + \mathbf{f} + \boldsymbol{\epsilon}, \quad (4)$$

where $\mathbf{f} = f(\mathbf{Z})$ is a vector of subject-specific random effects with $\mathbf{f} \sim (0, \tau \mathbf{K})$, and hence the null hypothesis of no association can be more simply expressed as $H_0 : \tau = 0$. This can be accomplished using a variance component score test with test statistic

$$Q = \frac{1}{2\hat{\sigma}_0^2} (\mathbf{y} - \hat{\mathbf{y}}_0)' \mathbf{K} (\mathbf{y} - \hat{\mathbf{y}}_0), \quad (5)$$

where $\hat{\mathbf{y}}_0$ is the predicted value of \mathbf{y} under the null model and $\hat{\sigma}_0^2$ indicates the estimated residual variance under the null model. Under H_0 , Q asymptotically follows a mixture of chi-square distributions,

$$Q \sim \sum_{i=1}^n \lambda_i \chi_{1,i}^2, \quad (6)$$

where $(\lambda_1, \dots, \lambda_n)$ are the eigenvalues of $P_0^{1/2} \mathbf{K} P_0^{1/2}$, where $P_0 = \mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the standard projection (hat) matrix, and $\chi_{1,i}^2$ are independent χ_1^2 random variables. However, due to the nature of microbiome kernels, a key difference between MiRKAT and kernel-based approaches for genetics data is that the kernels tend to be much more complicated and the sample sizes much smaller. Consequently, a small sample correction is used within MiRKAT, which is exact for quantitative y and approximate for dichotomous y .

MiRKAT's power is highest when the chosen kernel best represents the true form of association. To allow simultaneous testing with multiple kernel matrices, Optimal MiRKAT (OMiRKAT) uses as a test statistic $P_{\min} = \min(P_1, \dots, P_k)$, where each P_j is the corresponding P -value from a test for a single kernel as described above. Residual permutation is used to evaluate the significance of P_{\min} . The optimal test has power close to that of the “best” kernel choice. However, the choice of kernel does not affect the type I error of MiRKAT family tests, since the score test only requires fitting the model under the null hypothesis of no association.

As noted earlier, MiRKAT family of tests is based on kernel approaches designed for genetic association analysis (with some important modifications). Thus, an important aspect of these tests is that they can harness the rich, existing literature on kernel approaches for genetic association analysis which facilitates analysis

Table 3 MiRKAT P -values and R -squared values using unweighted (K_u), generalized ($K_{0.5}$), and weighted (K_w) UniFrac, Bray–Curtis (K_{bc}), and Sørensen–Dice kernels (K_s), as well as optimal MiRKAT

Outcome	Value	K_u	$K_{0.5}$	K_w	K_{bc}	K_s	Optimal
$\mathbf{y}_{\text{clust}}$	P	<0.001	0.004	0.057	0.007	<0.001	0.002
	R^2	0.008	0.004	0.002	0.003	0.005	-
$\mathbf{y}_{\text{common}}$	P	0.180	0.020	0.025	0.003	0.541	0.007
	R^2	0.005	0.004	0.001	0.003	0.003	-

of more complex end points, study designs, and analytic objectives. The reduced computational expense also makes the approach attractive for the purposes of power calculation via simulation and for screening large numbers of outcomes (where a large number of permutations would be necessary).

MiRKAT family analyses are available in the *R* package MiRKAT. The results from the MiRKAT analysis below are shown in Table 3.

MiRKAT

```
library(MiRKAT)
MiRKAT(y = y.clust, X = X, Ks = Ks, out_type = "C")
MiRKAT(y = y.common, X = X, Ks = Ks, out_type = "C")
```

As in the PERMANOVA analysis, the individual kernel most significantly associated with $\mathbf{y}_{\text{clust}}$ is unweighted UniFrac and Bray–Curtis; weighted UniFrac is the only nonsignificant kernel. However, in addition to individual kernel results, OMiRKAT indicates the overall significance of the association between the microbiome and $\mathbf{y}_{\text{clust}}$ considering all five kernels. An R^2 statistic is also reported. For continuous outcomes, the MiRKAT test statistic is proportional to the coefficient of determination in similarity matrix regression [40], but for most outcome types, it is recommended to use the MiRKAT R^2 as a relative measure to compare different kernels. In the Charlson data, MiRKAT R^2 shows that unweighted UniFrac explains more of the variability in $\mathbf{y}_{\text{clust}}$ than any of the other distances, as may be expected from the large number of phylogenetically clustered rare taxa associated with \mathbf{y} in that scenario. Moving to $\mathbf{y}_{\text{common}}$, again matching the PERMANOVA analysis, the Bray–Curtis dissimilarity is most significantly associated with this outcome. Although the unweighted UniFrac and Sørensen–Dice distances yield distinctively nonsignificant results, OMiRKAT still demonstrates strong evidence for an overall association.

4.3 Sum of Powered Score Tests

A third approach to distance-based testing falls under the umbrella of sum of powered score tests. The microbiome-based sum of powered score (MiSPU) tests address the problem of a high proportion of unassociated taxa leading to noise accumulation by adaptively weighting taxa based on importance, while still incorporating phylogenetic relationships [36]. We again assume a continuous outcome y for simplicity; MiSPU tests are available for continuous and binary outcome variables and are, like the previous tests, based on approaches developed for genetic studies.

Using the unweighted and weighted UniFrac distances as a starting point, generalized taxon proportions are defined as

$$Q_{ij}^u = b_j I(p_{ij} > 0), \quad Q_{ij}^w = b_j p_{ij}. \quad (7)$$

Then, a linear model is fit using these generalized taxon proportions,

$$y_i = \beta_0 + \boldsymbol{\beta}' \mathbf{X}_i + \sum_{j=1}^J \alpha_j Q_{ij} + \epsilon_i, \quad (8)$$

where ϵ_i is an error term with mean 0 and variance σ^2 . As with MiRKAT, the null hypothesis of no association between the microbiome and the outcome may be written as $H_0 : (\alpha_1, \dots, \alpha_J) = 0$. To test this null hypothesis, a score vector $\mathbf{U} = (U_1, \dots, U_J)$ is constructed via

$$U_j = \sum_{i=1}^n (y_i - \hat{y}_{i,0}) Q_{ij} \quad (9)$$

and used to construct the weighted score-based test statistic

$$T_{\text{MiSPU}(\gamma)} = \sum_{j=1}^J U_j^\gamma, \quad (10)$$

where $\gamma \geq 1$ is an integer that controls the weight placed on larger values of U compared to smaller values. Using the weighted generalized taxon proportions Q_{ij}^w leads to weighted MiSPU; unweighted generalized taxon proportions Q_{ij}^u correspond to unweighted MiSPU. P -values are calculated by residual permutation (see [36] for details).

The optimal choice of γ is primarily driven by the abundance and branch length of the associated taxa, as well as whether individual taxon effects are in the same or opposite directions. If associated taxa have short branch lengths b_j or small relative abundances p_{ij} , smaller values of γ will yield higher power by including these taxa in the aggregate statistic; if associated taxa are common and have long branch

Table 4 MiSPU and aMiSPU P -values. For unweighted and weighted MiSPU, the aSPU P -value is reported, with the γ value that resulted in the lowest individual P -value in parentheses

Outcome	Unweighted aSPU	Weighted aSPU	aMiSPU
$\mathbf{y}_{\text{clust}}$	$P = 0.003 (\hat{\gamma} = 2)$	$P = 0.291 (\hat{\gamma} = 2)$	$P = 0.005$
$\mathbf{y}_{\text{common}}$	$P = 0.352 (\hat{\gamma} = 2)$	$P = 0.106 (\hat{\gamma} = 3)$	$P = 0.217$

lengths, larger γ will result in higher power. Taking the limit as $\gamma \rightarrow \infty$ results in a test statistic that is proportional to $\max(|U_j|)$. When γ is even, taxa with effects in opposite directions all contribute to a larger sum statistic, whereas odd values of γ may allow taxa with positive and negative effects to cancel each other out and hence will result in a loss of power when taxa have opposite directional effects.

To prevent potential power loss due to poor choice of γ or weighted vs. unweighted tests, adaptive MiSPU (aMiSPU) uses as a test statistic the minimum P -value for multiple candidate values of γ , such as $\gamma = (2, 3, \dots, 8, \infty)$, and weighted and unweighted tests. Permutation is again used to assess the significance of P_{\min} .

The score statistics U_j may also be used to assess relative taxon importance via a contribution statistic $C_k = |U_k|^{\hat{\gamma}} / \sum_{j=1}^J |U_j|^{\hat{\gamma}}$, where $\hat{\gamma}$ and weighted versus unweighted U_k are chosen based on which resulted in the minimum P -value. These importance scores may also be used to select the top several taxa for interpretation and possibly further study.

Sample code for aMiSPU analysis is below, and results are shown in Table 4. As expected due to the intrinsic dependence on phylogenetic differences, aMiSPU provides strong evidence for an association between the microbiome and $\mathbf{y}_{\text{clust}}$, particularly due to the significance using the unweighted generalized taxon proportions. The γ value resulting in the lowest P -value is small in these analyses (2 or 3), consistent with placing similar weight on rare and common taxa (or taxa that are located in shallow and deep parts of the phylogenetic tree). For $\mathbf{y}_{\text{common}}$, however, since the associated taxa are not part of a phylogenetic cluster, aMiSPU does not detect an association.

aMiSPU

```
library(MiSPU)
MiSPU(y = y.clust, X = throat.otu.tab, tree = throat.
       tree, cov = X, model = "gaussian")
MiSPU(y = y.common, X = throat.otu.tab, tree = throat.
       tree, cov = X, model = "gaussian")
```

4.4 Adaptive Tests

Combining the kernel machine regression-based tests and the sum of powered score tests leads to adaptive association tests such as the optimal microbiome-based association test (OMiAT) [16] and optimal microbiome-based survival analysis (OMiSA) [17]. Combining classes of tests ideally retains the strengths of both classes, while balancing out weaknesses in each. Adaptive tests are available for continuous, binary, and censored time-to-event outcome variables; as said previously, we will focus on continuous outcomes and therefore on OMiAT. Though the adaptive sum of powered score (aSPU) test used in OMiAT is similar in spirit to aMiSPU, ordinary taxon proportions are used without weighting by branch length for this test.

The test statistic for OMiAT is

$$M_{\text{OMiAT}} = \min(T_{\text{aSPU}}, Q_{\text{OMiRKAT}}), \quad (11)$$

where T_{aSPU} is the minimum P -value statistic of the adaptive SPU test and Q_{OMiRKAT} is the minimum P -value statistic of the optimal MiRKAT test. Then, as in the previous two sections, the final P -value is computed by permutation, as described in [16].

An OMiAT analysis is displayed below, with results presented in Table 5. Consistent with previous results, OMiRKAT has a much lower P -value than aSPU for $\mathbf{y}_{\text{common}}$, though since OMiAT performs a non-phylogenetic SPU rather than using MiSPU, the severe reduction in power for non-phylogenetically based associations is not seen in aSPU. OMiAT's P -value is intermediate between OMiRKAT and aSPU in this setting. For $\mathbf{y}_{\text{clust}}$, aSPU and OMiAT show slightly more significant results than OMiRKAT.

OMiAT

```
## OMiAT is available on GitHub
#devtools::install_github("hk1785/OMiAT")
library(OMiAT)
library(ecodist) # OMiAT requires this
OMiAT(Y = y.clust, otu.tab = throat.otu.tab, tree =
      throat.tree, cov = X, model = "gaussian")
OMiAT(Y = y.common, otu.tab = throat.otu.tab, tree =
      throat.tree, cov = X, model = "gaussian")
```

Table 5 OMiAT P -values, as well as OMiAT-calculated aSPU and OMiRKAT P -values. OMiRKAT P -values are based on Bray–Curtis and unweighted, generalized ($\alpha = 0.5$), and weighted UniFrac kernels. aSPU is based on $\gamma \in \{1, 2, 3, 4, \infty\}$

Outcome	aSPU	OMiRKAT	OMiAT
y_{clust}	<0.001	0.001	<0.001
y_{common}	0.041	0.007	0.015

4.5 Comparison of Distance-Based Tests

The options for distance-based association testing presented above have a variety of strengths and weaknesses. In terms of modeling flexibility, the MiRKAT family allows the widest range of outcome types, whereas the other methods only permit continuous, binary, and perhaps survival outcomes. PERMANOVA, MiRKAT, and the adaptive tests allow consideration of multiple measures of distance or dissimilarity, which is vitally important given that using a dissimilarity that does not capture the true form of association can lead to drastic reductions in power. While aMiSPU allows the use of quantitative or qualitative data and, to some extent, control over the amount of weight placed on different depths of phylogeny or taxon abundance, the level of flexibility is lower.

In situations in which all four methods apply (i.e., continuous and binary outcomes), simulation results demonstrate that OMiAT nearly matches the better of OMiRKAT and aSPU in power across a range of association settings and, in some circumstances, has higher power than either aSPU or OMiRKAT alone [16]. In particular, OMiAT performs well when the true association comprises a mix of common and rare taxa that are not phylogenetically clustered. When a wide range of kernels are considered for OMiRKAT, it often matches and sometimes exceeds the power of OMiAT, with particularly high power when common taxa (regardless of direction of effect) or phylogenetically clustered taxa with the same direction of effect are associated with the outcome.

Comparing OMiRKAT and aMiSPU, the kernel machine regression tests have higher power when modest shifts in a larger number of taxa are associated with the outcome. The sum of powered score tests is stronger when many taxa are unassociated and a few have strong effects, particularly if the associated taxa have long branch lengths or high abundances.

In terms of computational efficiency, the exact computation of P -values in MiRKAT is much faster than PERMANOVA, aMiSPU, or OMiAT, all of which rely on permutation-based P -values. For small-to-moderate samples, this difference is unlikely to be the determining factor in choice of test, but for larger sample sizes, computational efficiency becomes increasingly important. This is also the case for simulation-based statistical power analysis and for screening large numbers of outcomes where very small P -values are needed due to the multiple testing thresholds.

Therefore, the choice of global test of association should be driven by the needs of a particular study, taking into consideration the sample size and computational resources available, the data type of the outcome of interest, and any prior knowledge of the expected form of association in order to maximize power.

5 Strengths, Weaknesses, and Future Directions

Distance-based analysis of microbiome beta diversity is a powerful tool for assessing global associations with the microbiome. Because the microbiome often includes hundreds to thousands of taxa and effect sizes are modest, the use of distance-based tests often results in higher power than testing associations with each individual taxon. Factors contributing to higher power are the aggregation of modest effect sizes, avoidance of correcting for multiple comparisons, and incorporation of many possible forms of association through the choice of distance or dissimilarity measures.

A key weakness of distance-based analysis is the lack of information about which individual taxa play a role in the association. Because optimal tests incorporate information from multiple distances that each emphasize different features of the microbiome, which distance has the lowest P -value may provide some insight into broad features of the association. For example, if unweighted UniFrac is most significant in PERMANOVA or OMiRKAT, the presence of rare taxa is likely driving the association with the outcome of interest. Conversely, if weighted UniFrac is most significant, clusters of common taxa are more likely to be driving the association. Similarly, in addition to the contribution statistic discussed above for aMiSPU, high significance of unweighted aMiSPU may indicate that presence of rare taxa is driving the association, whereas weighted MiSPU indicates that common taxa are likely more important. A large γ in aMiSPU suggests that more common taxa towards the bottom of the phylogenetic tree are important to the association.

To use OMiAT for association mapping, MiCAM (microbiome comprehensive association mapping) tests groups of taxa at each taxonomic level (phylum, class, etc.) for significance and applies multiple testing correction at each taxonomic rank [16]. Patterns of association can reveal both what taxonomic level(s) are most strongly associated with the outcome and which particular taxa at each level seem to be important contributors.

The effect size estimation is another challenge for distance-based analysis. Both PERMANOVA and MiRKAT provide R^2 measures attempting to quantify the relationship between the microbiome composition and variables of interest. Both provide valid measures; however, it is noteworthy that the magnitudes of the R^2 values vary dramatically. This reflects a general challenge in the field in which the notion of correlation is complex when considering a multivariate quantity. Accordingly, there are multiple types of correlation which are equally valid but different in terms of scale and mathematical basis. In addition, the correlation depends strongly on the choice of distance metric, and changing the distance can

result in very different estimates. Thus, some caution is needed in interpreting the provided R^2 values. Also, due to the difficulty of quantifying effect sizes, there is a risk of large sample sizes yielding statistical significance even though differences are quite small and would not be considered scientifically meaningful.

There is substantial recent interest in model-based approaches to community-level analysis of the microbiome, in which a fully specified joint model for all taxon abundances is specified and related to covariates or phenotypes using a multivariate generalized linear mixed model framework [22, 23]. Advantages of beta-diversity analyses such as those discussed in this section include the minimal assumptions necessary for valid inference the computational efficiency of many of these methods and the broad utility of beta diversity measures across different analysis approaches. In comparison, advantages of model-based approaches include explicit separation of mean and dispersion effects, formal evaluation of model fit, and a clearer path toward identification of individual taxon associations [32].

Development of association mapping and taxon selection tools within the context of distance-based analysis of microbiome beta diversity is ongoing. In addition, new distances are being proposed to accommodate additional features of microbiome data and more complex study designs; for example, pldist was recently introduced for longitudinal measures of dissimilarity and allows tests of whether changes in the microbiome across time are associated with an outcome [26]. New models and test statistics are also needed for complex study designs; continuing the example of longitudinal designs, a mixed model approach with variance component selection was recently proposed for distance-based longitudinal microbiome association studies [39]. Still more recent work has focused on another modern study design, namely, multi-omics studies, either cross-sectional or longitudinal. Continued innovation will be necessary for these and other complex modern designs for microbiome studies.

References

1. Aitchison, J., Greenacre, M.: Biplots of compositional data. *J. Roy. Stat. Soc. Ser. C* **51**, 375–392 (2002)
2. Aitchison, J., Barceló-Vidal, C., Martín-Fernández, J.A., Pawlowsky-Glahn, V.: Logratio analysis and compositional distance. *Math. Geol.* **32**(3), 271–275 (2000)
3. Anderson, M.J.: A new method for non-parametric multivariate analysis of variance. *Austral Ecol.* **26**, 32–46 (2001)
4. Anderson, M.J., Walsh, D.C.I.: PERMANOVA, ANOSIM, and the Mantel test in the face of heterogeneous dispersions: What null hypothesis are you testing? *Ecological Monographs* **83**(4), 557–574 (2013)
5. Anderson, M.J., Walsh, D.C., Clarke, R., Gorely, R.N., Guerra-Castro, E.: Some solutions to the multivariate Behrens-Fisher problem for dissimilarity-based analyses. *Aust. NZ. J. Stat.* **59**(1), 57–79 (2017)
6. Bray, J.R., Curtis, J.T.: An ordination of upland forest communities of southern Wisconsin. *Ecol. Monogr.* **27**, 325–349 (1957)

7. Chang, Q., Luan, Y., Sun, F.: Variance adjusted weighted UniFrac: a powerful beta diversity measure for comparing communities based on phylogeny. *BMC Bioinformatics* **12**(1), 118 (2011)
8. Charlson, E.S., Chen, J., Custers-Allen, R., Bittinger, K., Li, H., Sinha, R., Hwang, J., Bushman, F.D., Collman, R.G.: Disordered microbial communities in the upper respiratory tract of cigarette smokers. *PloS ONE* **5**(12), e15216 (2010)
9. Chen, J., Bittinger, K., Charlson, E.S., Hoffmann, C., Lewis, J., Wu, G.D., et al.: Associating microbiome composition with environmental covariates using generalized UniFrac distances. *Bioinformatics* **28**(16), 2106–2113 (2012)
10. Clarke, K.R.: Non-parametric multivariate analyses of changes in community structure. *Austral Ecol.* **18**(1), 117–143 (1993)
11. Clarke, K.R., Somerfield, P.J., Chapman, M.G.: On resemblance measures for ecological studies, including taxonomic dissimilarities and a zero-adjusted Bray-Curtis coefficient for denuded assemblages. *J. Exp. Mar. Biol. Ecol.* **330**(1), 55–80 (2006)
12. Fukuyama, J.: Emphasis on the deep or shallow parts of the tree provides a new characterization of phylogenetic distances. *Genome Biol.* **20**, 131 (2019)
13. Gabriel, K.R.: The biplot graphic display of matrices with application to principal component analysis. *Biometrika* **58**(3), 453–467 (1971)
14. Gloor, G.B., Macklaim, J.M., Pawlowsky-Glahn, V., Egozcue, J.J.: Microbiomem datasets are compositional: and this is not optional. *Front. Microbiol.* **8**, 2224 (2017)
15. Knight, R., Vrbanac, A., Taylor, B.C., Aksnenov, A., Callewaert, C., Debelius, J., Gonzalez, A., Koscielek, T., McCall, L.I., McDonald, D., Melnik, A.V., Morton, J.T., Navas, J., Quinn, R.A., Sanders, J.G., Swafford, A.D., Thompson, L.R., Tripathi, A., Xu, Z.Z., Zaneveld, J.R., Zhu, Q., Caporaso, J.G., Dorrestein, P.C.: Best practice for analysing microbiomes. *Nat. Rev. Microbiol.* **16**, 410–422 (2018)
16. Koh, H., Blaser, M.J., Li, H.: A powerful microbiome-based association test and a microbial taxa discovery framework for comprehensive association mapping. *Microbiome* **5**, 45 (2017)
17. Koh, H., Livanos, A.E., Blaser, M.J., Li, H.: A highly adaptive microbiome-based association test for survival traits. *BMC Genomics* **19**, 210 (2018)
18. Liu, D., Lin, X., Ghosh, D.: Semiparametric regression of multidimensional genetic pathway data: Least-squares kernel machines and linear mixed models. *Biometrics* **63**(4), 1079–1088 (2007)
19. Lozupone, C., Knight, R.: UniFrac: a new phylogenetic method for comparing microbial communities. *Appl. Environ. Microbiol.* **71**(12), 8228–8235 (2005)
20. Lozupone, C.A., Hamady, M., Kelley, S.T., Knight, R.: Quantitative and qualitative β diversity measures lead to different insights into factors that structure microbial communities. *Appl. Environ. Microbiol.* **73**(5), 1576–1585 (2007)
21. Martín-Fernández, J.-A., Hron, K., Templ, M., Filzmoser, P., and Palarea-Albaladejo, J.: Bayesian-multiplicative treatment of count zeros in compositional data sets. *Stat. Modelling* **15**(2), 134–158 (2015)
22. Niku, J., Warton, D.I., Hui, F.K.C., Taskinen, S.: Generalized linear latent variable models for multivariate count and biomass data in ecology. *J. Agric. Biol. Environ. Stat.* **22**, 498–522 (2017)
23. Ovaskainen, O., Abrego, N., Halme, P., Dunson, D.: Using latent variable models to identify large networks of species-to-species associations at different spatial scales. *Methods Ecol. Evol.* **7**(5), 549–555 (2016)
24. Pavine, S., Dufour, A.B., Chessel, D.: From dissimilarities among species to dissimilarities among communities: a double principal coordinate analysis. *J. Theor. Biol.* **228**, 523–537 (2004)
25. Plantinga, A., Zhan, X., Zhao, N., Chen, J., Jenq, R.R., Wu, M.C.: MiRKAT-S: a community-level test of association between the microbiota and survival times. *Microbiome* **5**(1), 17 (2017)
26. Plantinga, A.M., Chen, J., Jenq, R.R., Wu, M.C.: pldist: ecological dissimilarities for paired and longitudinal microbiome association analysis. *Bioinformatics* **35**(19), 3567–3575 (2019)

27. Shi, Y., Zhang, L., Do, K.A., Peterson, C.B., Jenq, R.: aPCoA: Covariate adjusted principal coordinates analysis. *Bioinformatics* (2020). <https://doi.org/10.1093/bioinformatics/btaa276>
28. Silverman, J.D., Washburne, A.D., Mukherjee, S., David, L.A.: A phylogenetic transform enhances analysis of compositional microbiota data. *eLife* (2017). <https://doi.org/10.7554/eLife.21887>
29. Sohn, M.B., Li, H.: A GLM-based latent variable ordination method for microbiome samples. *Biometrics* **74**, 448–457 (2018)
30. Tang, Z.Z., Chen, G., Alekseyenko, A.: PERMANOVA-S: association test for microbial community composition that accommodates confounders and multiple distances. *Bioinformatics* **32**(17), 2618–2625 (2016)
31. Tsilimigas, M.C.B., Fodor, A.A.: Compositional data analysis of the microbiome: fundamentals, tools, and challenges. *Ann. Epidemiol.* **26**(5), 330–335 (2016)
32. Warton, D.I., Blanchet, F.G., O’Hara, R.B., Ovaskainen, O., Taskinen, S., Walker, S.C., Hui, F.K.C.: So many variables: joint modeling in community ecology. *Trends Ecol. Evol.* **30**(12), 766–779 (2015)
33. Whittaker, R.H.: Vegetation of the Siskiyou mountains, Oregon and California. *Ecol. Monogr.* **30**(3), 279–338 (1960)
34. Whittaker, R.H.: Evolution and measurement of species diversity. *Taxon* **21**(2–3), 213–251 (1972)
35. Wong, R.G., Wu, J.R., Gloor, G.B.: Expanding the UniFrac toolbox. *PLoS ONE* **11**(9), e0161196 (2016)
36. Wu, C., Chen, J., Kim, J., Pan, W.: An adaptive association test for microbiome data. *Genome Med.* **8**, 56 (2016)
37. Xia, Y., Sun, J., Chen, D.G.: Statistical Analysis of Microbiome Data with R. Springer, Singapore (2018)
38. Xu, T., Demmer, R.T., Li, G.: Zero-inflated Poisson factor model with application to microbiome read counts. *Biometrics* (2020). <https://doi.org/10.1111/biom.13272>
39. Zhai, J., Kim, J., Knox, K.S., Twigg III, H.L., Zhou, H., Zhou, J.J.: Variance component selection with applications to microbiome taxonomic data. *Front. Microbiol.* **9**, 509 (2018)
40. Zhan, X.: Relationship between MiRKAT and coefficient of determination in similarity matrix regression. *Processes* **7**(2), 79 (2019)
41. Zhan, X., Plantinga, A., Zhao, N., Wu, M.C.: A fast small-sample kernel independence test for microbiome community-level association analysis. *Biometrics* **73**(4), 1453–1463 (2017a)
42. Zhan, X., Tong, X., Zhao, N., Maity, A., Wu, M.C., Chen, J.: A small-sample multivariate kernel machine test for microbiome association studies. *Genet. Epidemiol.* **41**(3), 210–220 (2017b)
43. Zhao, N., Chen, J., Carroll, I.M., Ringel-Kulka, T., Epstein, M.P., Zhou, H., Zhou, J.J., Ringel, Y., Li, H., Wu, M.C.: Testing in microbiome-profiling studies with MiRKAT, the microbiome regression-based kernel association test. *Am. J. Hum. Genet.* **96**(5), 797–807 (2015)

Part III

Statistical Models and Inference

Joint Models for Repeatedly Measured Compositional and Normally Distributed Outcomes



Ivonne Martin, Hae-Won Uh, and Jeanine Houwing-Duistermaat

1 Introduction

Biomedical studies often collect multiple outcomes from the same subject to reveal complex underlying biological mechanisms. It might be of interest to model the association between these outcomes and a set of covariates. A straightforward method is to use a multiple univariate regression model for each outcome separately with the other outcomes and covariates included as independent variables in the model. However, the randomness of the outcome variables needs to be modelled since ignoring this randomness may yield biased estimates of the parameters modelling the association between the outcomes [22]. Moreover, one might be interested in the association between a covariate and both outcomes simultaneously. A joint regression model is an approach for this purpose and also increases the statistical power to estimate the effects of covariates on outcomes by incorporating the correlation between observations from the same subject via random effects. However, this approach is challenging when the observations are from different types, for instance, a mixture of continuous and discrete outcomes. The reason is that a multivariate distribution of these outcomes cannot be formulated [6, 18].

I. Martin

Department of Epidemiology and Data Science, Amsterdam UMC, Amsterdam, The Netherlands
e-mail: i.martin@amsterdamumc.nl

H.-W. Uh

Department of Biostatistics and Research Support, UMC Utrecht, Utrecht, The Netherlands
e-mail: h.w.uh@umcutrecht.nl

J. Houwing-Duistermaat (✉)

Department of Statistics, Alan Turing Institute, University of Leeds, Leeds, United Kingdom
Department of Statistical Sciences, University of Bologna, Bologna, Italy
e-mail: j.duistermaat@leeds.ac.uk

Also, biomedical studies often have a cluster or a longitudinal design that induces additional correlation between observations from the same unit. In this chapter, we will focus on two outcomes, although the results could be generalized to more than two outcomes.

The presented methods are motivated by the repeated measurements of the gut microbial community and whole blood cytokine responses on subjects in the helminth-endemic area in Indonesia [16]. The goal of the analysis is to unravel the relationship between microbiome (MB) composition, immune response, and helminth infection. Here, helminth infection is the independent variable, while the MB composition and immune responses are the dependent variables or outcomes. The gut microbiome compositions are obtained from 16S rRNA gene sequencing. The processed data consists of counts of taxonomical data with a unit constraint for all taxonomical abundances with additional heterogeneity in the data due to measurement error or variability in the sampling of individuals. The observations on whole blood cytokine responses are continuous data representing the response of this cytokine to a specific antigen. Wammes et al. [25] have shown that helminth infection has an effect on whole blood cytokine responses and that this effect depends on treatment. Martin et al. [15] have shown a relationship between the MB and cytokine responses and that this relationship depends on the infection status. Here, a straightforward method was used where the cytokine responses are the outcomes, and infection, treatment, and MB composition expressed as a relative abundance for each bacteria taxon are the covariates. It was shown that the proportion of *Bacteroidetes* has a significant association with the interleukin-10 (IL-10) response to lipopolysaccharide (LPS) in uninfected subjects and that when the subjects were helminth-infected, the association between *Bacteroidetes* and IL-10 response to LPS is significantly different. This result suggests a role of helminth in changing the association between MB composition and cytokine responses; however, several limitations are noted. Firstly, the model assumes that the MB compositions are fixed, and hence, it does not account for the randomness due to measurement error. MB data obtained through 16S rRNA gene sequencing is observed with errors [21], adding an extra variation in the resulting data [20]. Furthermore, the joint effect of infection status on both outcomes cannot be assessed in this simple model. Thus, our objectives here are to characterize the association between covariates of interest and two outcomes and quantify the correlation between them.

Several works on the development of the joint model between continuous and discrete type outcomes in the biomedical research have been published, namely, between continuous and count data [10, 27], between continuous and time to event (reviewed in [19]), and continuous type with binary data [3, 4, 9], but less on multinomial type data. Here, we are dealing with the mixture of continuous and multivariate discrete outcomes with a constraint of the fixed total count. A review on formulating the joint model is given in [24].

When the objective is on modelling the association between covariates and two outcomes and quantification of the correlation between the two outcomes, shared random effects are used to account for the correlation between the multiple

outcomes from the same subject [6]. When the dataset has a complex correlation structure, several random effects are needed. In our motivating data, three types of correlation structures need to be modelled: the correlation between multiple categories at the same time, the correlation between the two outcomes simultaneously, and the correlations between multiple observations over time. A joint model requires two steps: firstly, formulating each submodel for the random variables, and secondly, modelling the relationship between these variables by introducing shared random effects. For each outcome, we consider a mixed model. For the multivariate counts, several distributions for a random effect modelling overdispersion have been proposed in the literature [12]. We have developed a mixed model for multivariate count data in which the overdispersion is modelled using the conjugate distribution and the correlation between observations at different time points is modelled by Gaussian random effects [17]. Here, we also propose to use a normally distributed random effect to model overdispersion and compare it with the combined method of conjugate and Gaussian distribution for the correlation between categories and within samples. The maximum likelihood approach [7] is used for parameter estimation and inference. Further, the marginal model is obtained by integrating over the random effect distribution using Gauss–Hermite quadrature.

The rest of this chapter is organized as follows. In Sect. 2, we describe the motivating dataset and the characteristics of samples. In Sect. 3, we present the proposed joint method in modelling the association of binary covariate with mixture types of outcomes. The code for the likelihood functions is provided. An evaluation of the proposed method’s performance in comparison with the naive method via simulations is described in Sect. 4. The results of applying the proposed method to the motivating dataset are given in Sect. 5, and we conclude and discuss the proposed method in Sect. 6. Section 7 presents all codes used in this chapter.

2 Motivating Data

The dataset considered here was measured in a subset of randomized controlled trials in a helminth–endemic area in Indonesia to assess the influence of helminth infection on whole blood cytokine responses as markers for human immune responses [26]. Households were randomized for a 400 mg albendazole or placebo. Treatments were administered once in every three months for one and a half years. Yearly stool samples were collected voluntarily, to detect the presence of helminth infections and obtain genomic material of the gut microbial community. Blood samples were drawn for immunological examinations. For the analyses, we used the observations at two different time points, namely, before treatment was commenced (pre-treatment) and 21 months after the first treatment (post-treatment).

Three different helminth species were observed. *Trichuris trichiura* infection was detected only by microscopy, while the DNA of hookworms (*Ancylostoma duodenale* and *Necator americanus*) and *Ascaris lumbricoides* was observed via multiplex real-time PCR. A subject who was infected with at least one helminth species was

Table 1 The characteristics of participants at pre-treatment

Characteristics	Albendazole ($N = 23$)	Placebo ($N = 39$)
Gender, female ($n(\%)$)	12 (52.17)	22 (56.41)
Age (mean(SD))	27.03 (15.80)	26.53 (15.86)
Helminth infections ($N(\%)$)		
Any helminths	16 (69.57)	23 (58.97)
Abundance of bacterial phyla, mean % (SD)		
<i>Firmicutes</i>	73.21 (10.76)	71.54 (12.94)
<i>Actinobacteria</i>	9.73 (5.84)	9.40 (7.75)
<i>Bacteroidetes</i>	6.70 (9.97)	7.27 (12.19)
Pooled	10.35 (7.29)	11.79 (8.10)
Cytokine responses (median, IQR)		
LPS	IL-10 250 (137.5, 400.5)	221 (137, 381.5)

regarded as helminth-infected. The pyrosequencing process of the 16S rRNA gene to obtain the bacterial data has been described in [15]. Here, we focus on two specific phyla, namely, *Bacteroidetes* and *Firmicutes*, and pool the remaining phyla into the pooled category. The blood cultures were stimulated to assess the innate and adaptive immune responses, characterized by cytokine. In [16], among all analyzed cytokine responses, only the innate interleukin (IL)-10 response to lipopolysaccharide (LPS) was significantly associated with *Bacteroidetes* proportion. In this analysis, we aim to reanalyze these outcomes simultaneously concerning helminth infections. Thus, we focus on the continuous type observation IL-10 response to LPS. Our data consists of 62 subjects who have complete measurements on MB composition and cytokine responses at pre- and post-treatment (Table 1).

3 Statistical Models

Let $Y_i^{(t)}$ be a continuous random variable observed for subject i , $i = 1, \dots, N$ at time point t , $t = 1, \dots, T$, and let $\mathbf{C}_i^{(t)} = \{C_{i1}^{(t)}, \dots, C_{iJ}^{(t)}\}$ be a J -dimensional vector of random variables of multivariate counts with a fixed total count $C_{i+}^{(t)}$ (compositional data). These counts are observed with an error that results in an additional source of variation. Let $\mathbf{X}_i^{(t)}$ be a vector of covariate values for subject i at time point t , which may influence both outcome variables $Y_i^{(t)}$ and $\mathbf{C}_i^{(t)}$. In addition, we assume that $Y_i^{(t)}$ and $\mathbf{C}_i^{(t)}$ are both influenced by an unobserved latent variable that results that these two variables are correlated. Our aim is to model the variable Y and assess its relationship with the random variable C and the covariate X while taking into account the presence of measurement error in the multivariate counts and the effect of the covariate X on C .

Martin et al. [16] used a simple linear mixed model to assess the relationship between the continuous outcome Y_i and the proportion of counts of category j , $\frac{C_{ij}}{C_{i+}} = \pi_{ij}$ in the longitudinal setting. Specifically, the following model was used:

$$Y_i^{(t)} = \mathbf{X}_i^{(t)} \boldsymbol{\xi} + \gamma_j \pi_{ij}^{(t)} + u_i + \epsilon_i^{(t)}. \quad (1)$$

Here, γ_j represents the association between the proportion of counts in category j and variable Y . The random subject-specific effect u_i represents the deviation of the population mean for subject i . This model ignores the fact that the multivariate count data are subject to measurement error, which may result in biased estimate of the parameter representing the association between the two outcome variables [22]. Further, the model only includes one category and hence does not assess the association between the vectors C_i and Y . Including all categories in the model is not straightforward due to the compositional nature of the data, which leads to the collinearity of the categories. To address these issues, we propose using joint models, i.e., the two submodels for Y and C are linked via shared random effects. Conditional on these random effects and the covariate X , the two variables Y and C are assumed to be independent.

We will first consider models for the counts and the covariates and then add a shared effect to link the two submodels for the counts C and the outcome Y . When modelling the association between a categorical count variable and a second categorical variable Z_k with K categories, the log-linear model is commonly used [1, 23]. Specifically, we follow the formulation of a saturated log-linear model in [17] where multivariate count outcome \mathbf{C} is represented as variable E and categorical variable Z as variable F ,

$$\begin{aligned} \log(\mu_{jk}^{(t)}) &= (\lambda_0 + \lambda_j^E) + (\lambda_k^F + \lambda_{jk}^{EF}) \\ &= \xi_{0j} + \xi_{1jk}[Z^{(t)} = k], \quad 1, \dots, J, \quad k = 1, \dots, K. \end{aligned} \quad (2)$$

Here, $\mu_{jk}^{(t)} = E(C_{jk}^{(t)})$ and $[\cdot]$ as the indicator variable. Identifiability of the parameters is obtained by imposing constraints. This model will be extended to account for extra variation due to measurement error (overdispersion), the correlation between observations at the various time points t , and the correlation with Y . We will consider two approaches: the mixed-effect multinomial logistic model [8], where all extra variations are modelled by normally distributed random effects, and the mixed-effect Dirichlet-multinomial of Martin et al. [17], where the overdispersion is modelled with the conjugate distribution and the other random sources are modelled with normally distributed random effects (combined likelihood). For the continuous variable Y , a linear mixed model with a random intercept u_Y [11] will be used for both approaches. The goodness of fit of the two proposed models will be assessed by comparing the observed correlation with the modelled marginal correlations.

3.1 The Multinomial Logistic Mixed Model (MLMM)

The multinomial logistic model achieves identifiability by assigning one category to be the baseline. This is typically one of the most common categories. Without loss of generality, we assume that the first category is the baseline category. Following the generalized linear model framework, the association between the proportion of counts in the j th category ($j = 2, \dots, J$) and the covariates \mathbf{X} is modelled as follows:

$$\text{logit} \left(\frac{\pi_{ij}^{(t)}}{\pi_{i1}^{(t)}} \right) = \xi_{0j}^{(t)} + \mathbf{X}_{ij}^{(t)} \boldsymbol{\xi}_j. \quad (3)$$

Now let the random effect u_{ij}^C represent the shared effects among the time points for category j of subject i . We assume that the measurement error does not change over time and is also represented by u_{ij}^C .

The corresponding regression model is defined as follows.

$$\text{logit} \left(\frac{\pi_{ij}^{(t)}}{\pi_{i1}^{(t)}} \right) = \xi_{0j}^{(t)} + \mathbf{X}_i^{(t)} \boldsymbol{\xi}_j^C + u_{ij}^C, \quad j = 2, \dots, J, \quad (4)$$

with the first category as a reference. The random effects $\mathbf{u}_i^C = \{u_{i2}^C, \dots, u_{iJ}^C\}$ follow a multivariate normal distribution with zero mean and the following symmetric covariance matrix Σ^C :

$$\Sigma^C = \begin{pmatrix} \sigma_{u_{C_2}}^2 & & & & \\ \rho \sigma_{u_{C_2}} \sigma_{u_{C_3}} & \sigma_{u_{C_3}}^2 & & & \\ \vdots & \vdots & \ddots & & \\ \rho \sigma_{u_{C_2}} \sigma_{u_{C_J}} & \rho \sigma_{u_{C_3}} \sigma_{u_{C_J}} & \dots & \sigma_{u_{C_J}}^2 & \end{pmatrix}.$$

Finally, when modelling both outcomes simultaneously, we need to introduce additional random effects to model the association between the two types of outcomes, namely, the random shared effect $\mathbf{U}_i^{(S)}$. Thus, for three categories, the joint model is as follows:

$$\begin{aligned} \log \left(\frac{\pi_{2i}^{(t)}}{\pi_{1i}^{(t)}} \right) &= \xi_{02}^{(t)} + \mathbf{X}_i^{(t)} \boldsymbol{\xi}_2^C + u_{i2}^C + u_{i2}^S \\ \log \left(\frac{\pi_{3i}^{(t)}}{\pi_{1i}^{(t)}} \right) &= \xi_{03}^{(t)} + \mathbf{X}_i^{(t)} \boldsymbol{\xi}_3^C + u_{i3}^C + u_{i3}^S \\ Y_i^{(t)} &= \xi_0^{(t)} + \mathbf{X}_i^{(t)} \boldsymbol{\xi}_Y + u_{i2}^S + u_{i3}^S + u_y + \epsilon_i^{(t)}. \end{aligned} \quad (5)$$

Thus, a vector of random effects \mathbf{u}_i^* can be defined as follows:

$$\mathbf{u}_i^* = \begin{pmatrix} u_{i2}^C + u_{i2}^S \\ u_{i3}^C + u_{i3}^S \\ u_{i2}^S + u_{i3}^S + u_y \end{pmatrix} \sim \text{MVN}(\mathbf{0}_3, \Sigma_M),$$

with

$$\Sigma_M = \begin{pmatrix} \sigma_{u_{C2}}^2 + \sigma_{u_{S2}}^2 & \rho\sigma_{u_{C2}}\sigma_{u_{C3}} & \sigma_{u_{S2}}^2 \\ \rho\sigma_{u_{C2}}\sigma_{u_{C3}} & \sigma_{u_{C3}}^2 + \sigma_{u_{S3}}^2 & \sigma_{u_{S3}}^2 \\ \sigma_{u_{S2}}^2 & \sigma_{u_{S3}}^2 & \sigma_{u_{S2}}^2 + \sigma_{u_{S3}}^2 + u_y^2 \end{pmatrix}. \quad (6)$$

This model is the joint model with the multinomial logistics mixed model as a submodel for the counts (MLMM).

Estimates of all parameters are obtained by maximizing the likelihood of the joint distribution (7). The joint marginal distribution is

$$\Pr(\mathbf{C}_i, \mathbf{Y}_i) = \int \Pr(\mathbf{C}_i, \mathbf{Y}_i | \mathbf{U}_i^S) \Pr(\mathbf{U}_i^S) d\mathbf{U}_i^S. \quad (7)$$

Since the likelihood does not have a closed-form formula, numerical approximations, such as Gauss–Hermite quadrature, need to be utilized. To compute the marginal distribution, we followed the integration of the multivariate Gauss–Hermite quadrature described in Liu and Pierce [13]. For the code, we need the following definitions. Let ff be the integrand of $\Pr(\mathbf{C}_i, \mathbf{Y}_i | \mathbf{U}_i^S) \Pr(\mathbf{U}_i^S)$ given in equation (7). Let \mathbf{Yt} be the observation for one subject, FixEf be the function to define the $\log(E(\mu_j))$ for each j , Des the design matrix, \mathbf{b} the vector of parameters, CNF the marginal distribution for multivariate count outcome, and \mathbf{z} be the vector of random-effect parameters. Now the code for the function ff is as follows.

R code: computing the marginal distribution for joint method with MLMM

```

eta1 <- FixEf(Yt[7], b[1:4], Des, method)
eta2 <- FixEf(Yt[8], b[1:4], Des, method)
eta1 <- exp(eta1 + z[1:2])
eta2 <- exp(eta2 + z[1:2])
Eta1 <- c(1,eta1)
Eta2 <- c(1,eta2)
ff <- function(z){CNF(Yt[1:3], Eta1, method) +
    CNF(Yt[4:6], Eta2, method) +
    dnorm(Yt[9],mean=c(1,Yt[7])%*%b[5:6] + z[3],sd=ev,log=TRUE) +
    dnorm(Yt[10],mean=c(1,Yt[8])%*%b[5:6] + z[3],sd=ev,log=TRUE) +
    dmvnorm(z, mean = rep(0,3), sigma = Sigma, log=TRUE) }
```

The mode and variance for our integrand can be obtained by using the following code:

```
opt <- try(optim(c(0.1,-0.2,0.1),ff,method="BFGS",
control=list(fnscale = -1,maxit=9000),hessian=TRUE)) .
```

The output `opt` contains `opt$par` that is the mode of `ff` and `-opt$hessian` that is the Fisher information matrix at this mode.

Now, the log-likelihood (7) can be approximated by multivariate Gauss–Hermite quadrature. To approximate the integral of

$$\int_{-\infty}^{\infty} \exp(-x^2) h(x) dx \approx \sum_{i=1}^{n_{GQ}} w_i h(x_i),$$

where the integrand has mode 0 and variance 1, and R has a built-in function `gauss.hermite` from the package `ecoreg`. This function returns a matrix of 2 columns, consisting of points (x_i) and weights (w_i). However, the mode and variance of our integrand are not around 0 and 1, respectively. Therefore, these weights and points need to be transformed. In the code below, we calculate `a.star`, which is a set of transformed points. Then, `y`, which is the vector of `ff` values at these transformed points, can be computed. Using `y` and the transformed weights $w_i \exp(x_i^2)$, the likelihood `L` can be computed using the following code.

```
FIM <- nearPD(-opt$hessian)$mat
invDer <- solve(FIM)
CH <- chol(invDer)
a.star <- t(opt$par + (sqrt(2)*CH%*%t(x)))
y <- apply(a.star,1,ff)
L <- (2^(3/2))*sqrt(det(CH))*sum(exp(y)*w*exp(x^2)) }
```

For more details about this formulation, see Liu and Pierce [13]. For the complete code to compute the likelihood, see Sect. 7 under the function `integralComputation`.

An alternative to using the normal distribution for the overdispersion is to use the conjugate distribution. Combining this model with normally distributed effects yields the Dirichlet-multinomial mixed model (see [5]), which will be described in the next section.

3.2 Dirichlet-Multinomial Mixed Model (DMMM)

Martin et al. [17] used the combined model where a random effect with the conjugate distribution models the overdispersion and a normally distributed random effect in the linear predictor models the correlation over t . This is the Dirichlet-multinomial mixed model. For each subject i , the count vector \mathbf{C}_i is measured at time points t . A Gaussian random subject effect u_i is introduced to account for the correlation within a subject between time points. Specifically, the model is

$$\begin{aligned} \left\{ \frac{C_{i1}^{(t)}}{C_{i+}^{(t)}}, \dots, \frac{C_{i1}^{(t)}}{C_{i+}^{(t)}} \right\} \mid \left\{ \eta_{i1}^{(t)}, \dots, \eta_{ij}^{(t)} \right\}, u_i \sim \text{Mult} \left(\tilde{\pi}_{i1}^{(t)}, \dots, \tilde{\pi}_{iJ}^{(t)} \right), \\ \left\{ \tilde{\pi}_{i1}^{(t)}, \dots, \tilde{\pi}_{iJ}^{(t)} \right\} \sim \text{Dir} \left(\eta_{i1}^{(t)}, \dots, \eta_{iJ}^{(t)} \right), \\ \eta_{ij}^{(t)} = \theta^{-1} \mu_{ij}^{(t)}, \end{aligned} \quad (8)$$

with μ representing the regression model with the covariate. Here, the conjugate distribution models the correlation among the categories for one subject at a specific time point. The correlation between the same categories at different time points is modelled by normally distributed random effects. Hence,

$$\log \left(\mu_{ij}^{(t)} \right) = (\lambda_0 + \lambda_j^E) + (\lambda_k^F + \lambda_{jk}^{EF}) [X_i^{(t)} = k] + u_{ij}^C, \quad (9)$$

with baseline constraints (i.e., $\lambda_1^E = \lambda_1^F = \lambda_{1k}^{EF} = \lambda_{j1}^{EF} = 0$) and where the vector of random effects $\mathbf{u}_i^C = \{u_{i1}^C, u_{i2}^C, \dots, u_{iJ}^C\} \sim \text{MVN}(0, \Sigma)$ with

$$\Sigma = \begin{pmatrix} \sigma_{u_{C_1}}^2 & 0 & 0 & 0 \\ 0 & \sigma_{u_{C_2}}^2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma_{u_{C_J}}^2 \end{pmatrix}$$

Identifiability is obtained by assuming $\theta^{-1} \exp(\lambda_0) = \delta_0^{-1}$. Parameters $\lambda_{jk}^{EF}, j = 2, \dots, J$ and $k = 1$ have the same interpretation as $\xi_j^C, j = 2, \dots, J$ in MLMM.

For the continuous outcome variable Y , we again use a linear mixed-effect model with one subject-specific random effect u_Y [11]. For the joint model, we introduce the additional random effects to model the association between the two types of outcomes, namely the random shared effect $\mathbf{U}_i^{(S)}$. Thus, for three categories, the joint model is as follows. In the case that multivariate bacterial counts were assumed to follow the Dirichlet-multinomial mixed model (DMMM), the joint method is formulated as follows:

$$\begin{aligned}
\log(\mu_{i1}^{(t)}) &= \lambda_0 + \lambda_2^F [X_i^{(t)} = k] + u_{i1}^C + u_{i1}^S, \\
\log(\mu_{i2}^{(t)}) &= (\lambda_0 + \lambda_2^E) + (\lambda_2^F + \lambda_{22}^{EF}) [X_i^{(t)} = k] + u_{i2}^C + u_{i2}^S, \\
\log(\mu_{i3}^{(t)}) &= (\lambda_0 + \lambda_3^E) + (\lambda_2^F + \lambda_{32}^{EF}) [X_i^{(t)} = k] + u_{i3}^C + u_{i3}^S, \\
Y_i^{(t)} &= \xi_0^{(t)} + \mathbf{X}_i^{(t)} \boldsymbol{\xi}^Y + u_{i1}^S + u_{i2}^S + u_{i3}^S + u_y + \epsilon_i^{(t)}. \tag{10}
\end{aligned}$$

Now the random effect u_i^* is defined as follows:

$$\mathbf{u}_i^* = \begin{pmatrix} u_{i1}^C + u_{i1}^S \\ u_{i2}^C + u_{i2}^S \\ u_{i3}^C + u_{i3}^S \\ u_{i1}^S + u_{i2}^S + u_{i3}^S + u_y \end{pmatrix} \sim \text{MVN}(\mathbf{0}_4, \Sigma_D),$$

$$\Sigma_D = \begin{pmatrix} \sigma_{u_{C1}}^2 + \sigma_{u_{S1}}^2 & 0 & 0 & \sigma_{u_{S1}}^2 \\ 0 & \sigma_{u_{C2}}^2 + \sigma_{u_{S2}}^2 & 0 & \sigma_{u_{S2}}^2 \\ 0 & 0 & \sigma_{u_{C3}}^2 + \sigma_{u_{S3}}^2 & \sigma_{u_{S3}}^2 \\ \sigma_{u_{S1}}^2 & \sigma_{u_{S2}}^2 & \sigma_{u_{S3}}^2 & \sigma_{u_{S1}}^2 + \sigma_{u_{S2}}^2 + \sigma_{u_{S3}}^2 + \sigma_{u_Y}^2 \end{pmatrix} \tag{11}$$

The marginal distribution for the joint model with DMMM can be formulated in the same way as in equation (7).

R code: computing the marginal distribution of joint method with DMMM

Let `ff` be the integrand of (7) where the distribution of multivariate count is DMMM. The procedure is very similar to the R code to compute the marginal distribution of joint method with MLMM. Note that the dimension of the covariance structure is 4 by 4 instead of 3 by 3, as is the case for MLMM.

```

eta1 <- FixEf(Yt[7], b[1:6], Des, method)
eta2 <- FixEf(Yt[8], b[1:6], Des, method)
eta1 <- (1/theta) * exp(eta1 + c(z[1:3]))
eta2 <- (1/theta) * exp(eta2 + c(z[1:3]))
ff <- function(z) { }CNF(Yt[1:3], eta1, method) +
CNF(Yt[4:6], eta2, method) +
dnorm(Yt[9], mean=c(1, Yt[7]) %*% b[7:8] + z[4], sd =ev, log=TRUE) +
dnorm(Yt[10], mean=c(1, Yt[8]) %*% b[7:8] + z[4], sd=ev, log=TRUE) +
dmvnorm(z, mean = rep(0,4), sigma = Sigma, log=TRUE) }

opt <- try(optim(c(0.1, 0.1, -0.2, 0.1), ff, method="BFGS",

```

```

control=list(fnscale = -1,maxit=9000),hessian=TRUE)
FIM <- nearPD(-opt$hessian)$mat
invDer <- solve(FIM)
CH <- chol(invDer)
a.star <- t(opt$par + (sqrt(2)*CH%*%t(x)))
y <- apply(a.star,1,ff)
L <- (2^(4/2))*sqrt(det(CH))*sum(exp(y)*w*exp(x^2))

```

3.3 Goodness of Fit

The two models yield a different correlation structure among the categories and the continuous outcome. To assess the fit of the models, the observed correlations might be compared to the modelled correlations. The variances of the shared effects u_S represent the association between the two types of outcome and are often of interest. However, these variances may be hard to interpret as correlations and are not directly observed. Therefore, we propose to compute the marginal correlation of the model and compare these with the observed correlations:

$$\text{Corr}\left(C_{ij}^t, Y_i^t\right) = \frac{\sigma_{C_{ij}^t, Y_i^t}}{\sqrt{\sigma_{C_{ij}^t}^2 \sigma_{Y_i^t}^2}}.$$

To compute the first and second moments of the marginal distributions for the models, Monte Carlo sampling might be used.

4 Simulation Studies

A set of simulation studies were conducted with the following two objectives. Firstly, we investigate the performances of the proposed joint methods compared to the aforementioned naive method. Secondly, the performance of the two joint methods in estimating the marginal covariance structure among the outcomes is evaluated. For the first objective, we are especially interested in the model for the continuous outcome, the estimator's performance for the covariate effects, and the standard deviations of the shared effects. For the second objective, we compared the observed marginal correlation with the estimated marginal correlation.

With regard to the random-effect structure for the joint models, we consider a category-dependent shared random effect for the joint model with DMMM and a logit-dependent random effect for the joint model with MLMM, and we assume different standard deviations for each category or logit. The simulation study was performed in R statistical software. The computation of the Gauss–Hermite integral

is given in Sect. 7 under the function of `integralcomputation`. Here, three knots of Gauss–Hermite quadrature are used.

4.1 Simulation Setting

We generated counts for $N = 50$ subjects, and the total count for the multivariate outcome was $C_{i+} = 2000$. For the parameters modelling the relationship between the fixed effects and the two outcome variables, we based the values on the estimates obtained in the data analysis. These parameters were the same for the DMMM and the MLMM, except that DMMM has one more parameter for the model of the multivariate counts.

Specifically for the joint model with MLMM, the fixed-effect parameters were as follows:

$$\boldsymbol{\xi} = \left\{ \xi_{02}^C, \xi_{12}^C, \xi_{03}^C, \xi_{13}^C, \xi_0^Y, \xi_1^Y \right\} = \{-3.5, 0.8, -1.3, -0.15, -2.3, 0.1\}.$$

These parameters represent the intercepts and covariate effects for continuous outcome (ξ_0^Y, ξ_1^Y) and for each category's logits $(\xi_{02}^C, \xi_{12}^C, \xi_{03}^C, \xi_{13}^C)$. The standard deviations of the random effects were $\{\sigma_{u_{C_2}}, \sigma_{u_{C_3}}, \sigma_{u_Y}, \sigma_\epsilon\} = \{1, 0.8, 0.9, 0.7\}$, and the correlation coefficient between the measurement errors was $\rho = 0.1$. For the standard deviations of the shared random effects, we considered two sets of values namely $\{\sigma_{u_{S_2}}, \sigma_{u_{S_3}}\} = \{(0.5, 0.6), (1, 0.9)\}$, which later on will be labelled as low- and high-level variance, respectively.

We used the following procedure to generate datasets under this joint method with MLMM:

1. Based on the standard deviations of the random effects, we generated a multivariate normal random effect \mathbf{u}_i^* with covariance matrix Σ as defined in equation (6).
2. Based on the fixed-effects parameters, and by using the parameterization of the conditional mean given in (5), we generated the normally distributed and multinomial count outcomes for a subject.

For the joint model with DMMM, the following parameters for fixed effects were used: $\boldsymbol{\xi} = \{\lambda_2^F, \lambda_2^E, \lambda_3^E, \lambda_{22}^{EF}, \lambda_{32}^{EF}, \xi_0^Y, \xi_1^Y\} = \{0.5, -3.5, -1.3, 0.8, -0.15, -2.3, 0.1\}$. The standard deviations of the random effects modelling the covariance structure of the counts and the continuous outcomes were: $\{\sigma_{u_{C_1}}, \sigma_{u_{C_2}}, \sigma_{u_{C_3}}, \sigma_{u_Y}, \sigma_\epsilon\} = \{1, 1, 0.9, 0.9, 0.7\}$, which will be labelled as high- and low-level variance, respectively. An overdispersion parameter was set to $\theta = 0.1$.

We used the following procedure to generate datasets for the joint method with DMMM:

1. Based on the standard deviations of the random effects, we generated a multivariate normal random effect \mathbf{u}_i^* with covariance matrix Σ as defined in equation (11).

2. Based on the fixed-effects parameters and by using the parameterization of the conditional mean given in (10), we generated the normally distributed and multinomial count outcomes for a subject using equation (8).

Finally, for computing the marginal correlation, we generated 10,000 replicates, each with a sample size of 500 following the above procedure from each joint model. For each replicate, we compute the observed marginal correlation, taking the average of all datasets, and compare it with the estimated marginal correlation obtained from Monte-Carlo simulation.

4.2 Simulation Results

For both models, the Dirichlet-multinomial mixed model and the multinomial logistic mixed model, the estimated fixed-effect parameters of interest ξ_1^Y were unbiased and more efficient than when using the naive approach (Figs. 1 and 2). This holds for $\xi_1^Y = 0.1$ and $\xi_1^Y = 1$. For the estimated fixed-effect parameters of the second and third bacterial categories, both joint methods produced unbiased estimates (Fig. 5a and b, Appendix).

Concerning the estimation of the standard deviations of the shared effects of the second and third categories, the joint method with MLMM produced unbiased estimates, while for the joint method with DMMM, these were slightly overestimated (Figs. 3 and 4). For the standard deviations of the random effects in the model for the continuous outcome, the joint method with MLMM gave unbiased estimates

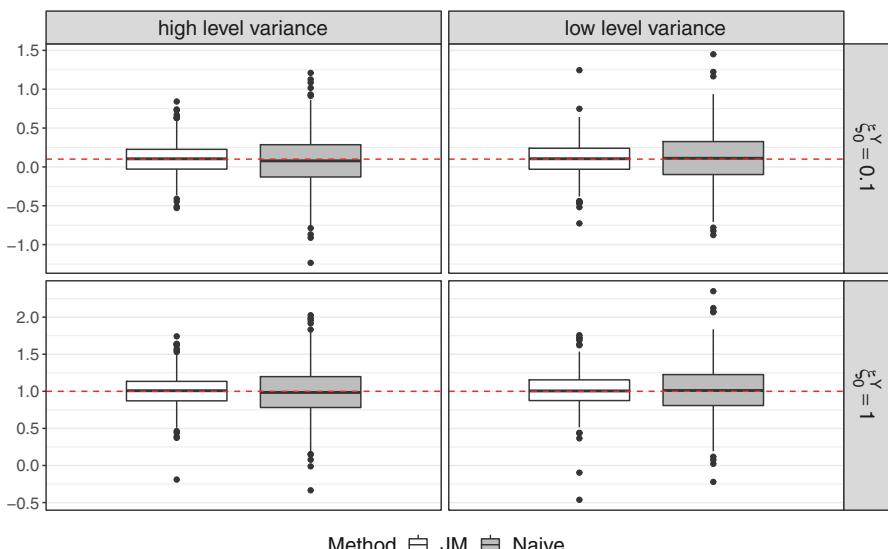


Fig. 1 Simulation result: the point estimates of the covariate of interest from joint method with Dirichlet-multinomial mixed model and naive approach

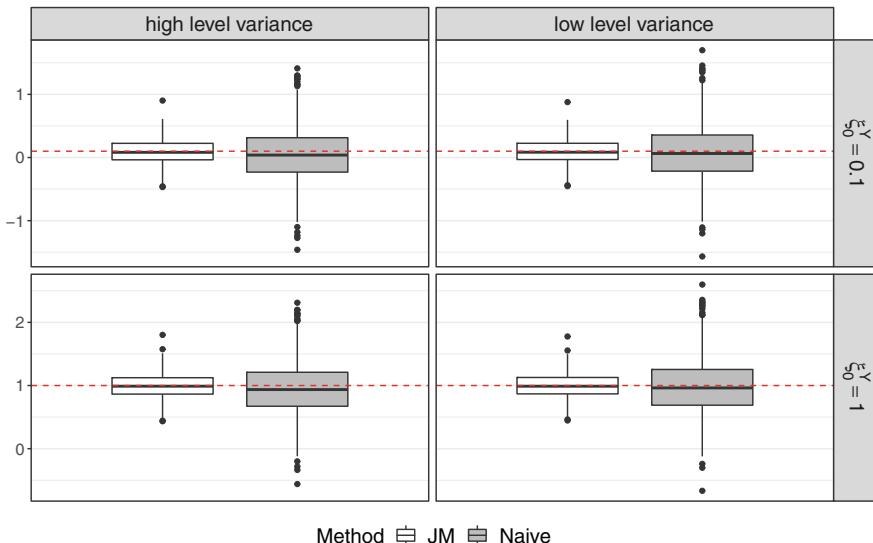


Fig. 2 Simulation result: the point estimates of the covariate of interest from joint method with multinomial logistics mixed model and naive approach

for $\log(\sigma_{uY})$ (Fig. 6, Appendix), although some outliers were observed. The joint model with DMMM appeared to underestimate $\log(\sigma_{uY})$ (Fig. 6, Appendix).

Finally, we compared the performance of each joint method in estimating the marginal correlation. For this purpose, we compared the observed and estimated marginal correlations within each joint method's replicates. For both joint methods, the marginal correlation appeared to closely resemble the observed marginal correlation from their corresponding joint method (Tables 8, 9, 10, 11).

5 Data Analysis

To assess the relationship between the IL-10 response and the MB composition, we first applied the naive approach using all data in the longitudinal setting. The estimated parameters are given in Table 2. The association between helminth infections and IL-10 response is not significant, but the associations between *Bacteroidetes* proportion and IL-10 to LPS are significantly different depending on the infection status. When subjects were helminth-uninfected, the cytokine responses and *Bacteroidetes* proportion are negatively associated, while this association disappears when subjects were helminth-infected. These findings suggest that MB composition is likely to correlate with cytokine response. The following code is used to estimate the parameters from the naive model.

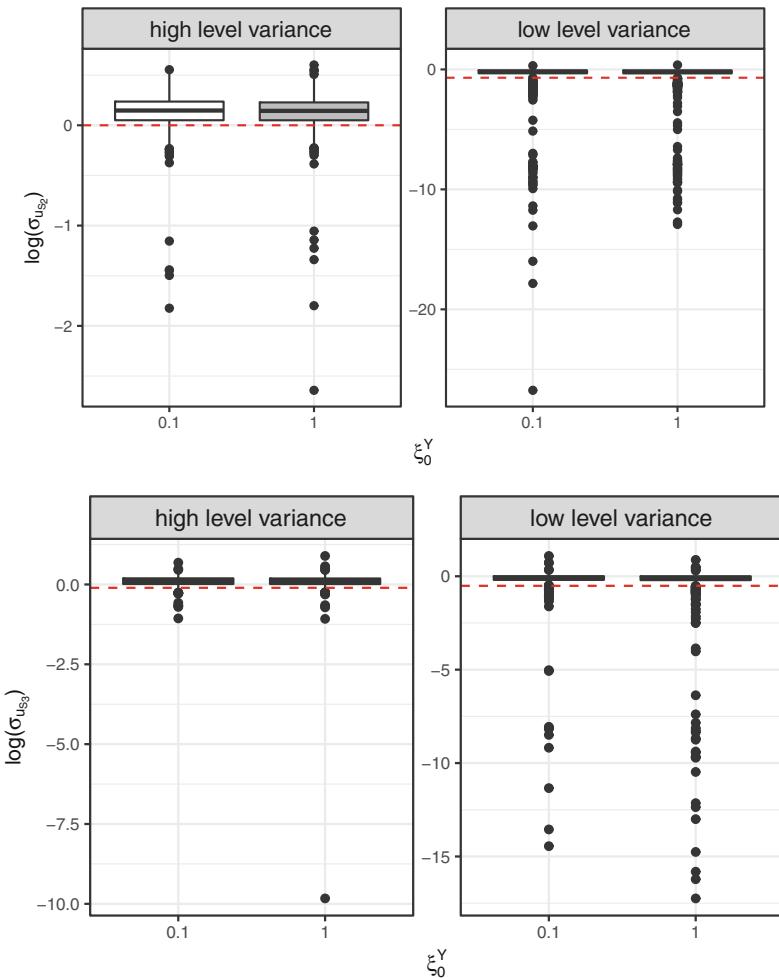


Fig. 3 The estimated standard deviation for the shared effects of the second and third categories obtained from joint methods with DMMM. The upper plots are for the variability of the shared effect for the second category, and the lower plots are for the third category

R code for fitting the naive model

```
library(lme4)
fit <- lmer(y ~ inf + inf*p.Firmi + inf*p.Bactero
+ (1|ID), REML = FALSE, data = bact)
```

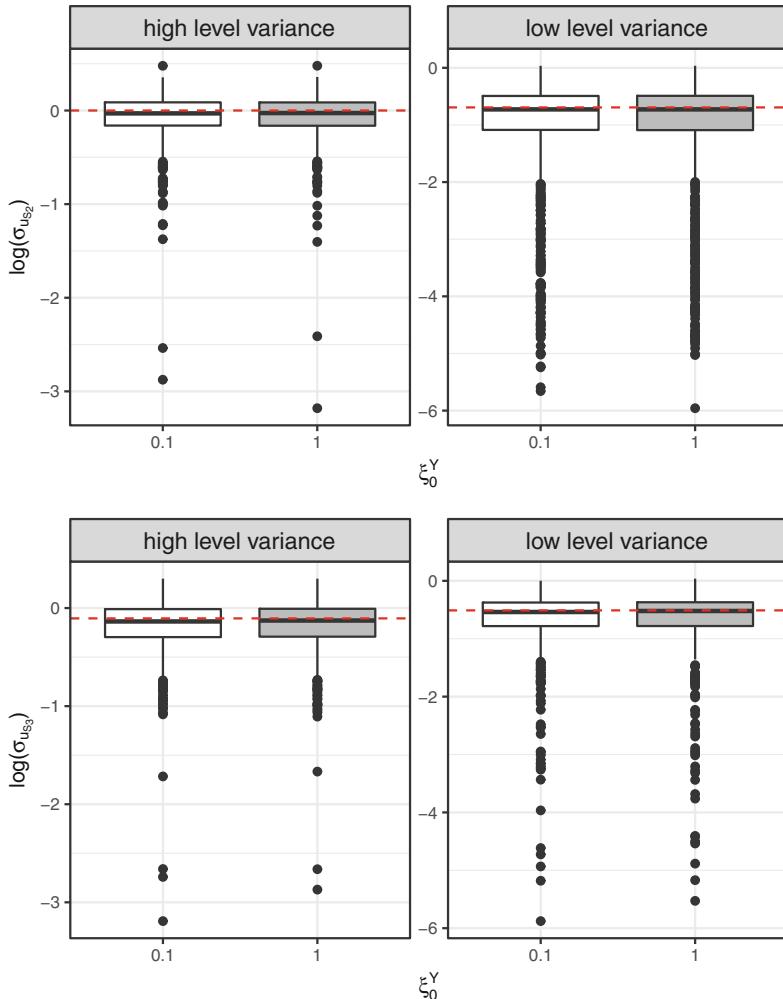


Fig. 4 The estimated standard deviation for the second and third categories from the joint method with MLMM. The upper plots are for the second logit, and the lower plots are for the third logit

The naive approach ignores the measurement error of the multivariate counts and the fact that the covariate may affect both outcomes. The joint methods that are proposed in this chapter might be more appropriate. However, the R code used in the simulation study did not work well for the data application. Instead of R, the SAS software with proc NLMIXED was used.

We considered the two approaches: the joint method with DMMM and with MLMM. For the joint method with MLMM, we used model (5) with the infection status as covariate and a random effect $\mathbf{u}_i^* = \{u_{C_2} + u_{S_2}, u_{C_3} + u_{S_3}, u_{S_2} + u_{S_3} + u_Y\}$ following a multivariate normal distribution with a mean of zero and covariance

Table 2 Data analysis: The estimates of the fixed-effect and random-effect parameters from the naive approach

Parameters	Fixed effects		Random effects	
	Estimate (s.e)	p-values	Group name	Variance
(Intercept)	2.33 (0.28)	<0.001	Individual	0.03
tpoint	-0.20 (0.06)	<0.001	Residual	0.09
inf	-0.09 (0.39)	0.82		
p.Firmi	0.11 (0.37)	0.77		
p.Bactero	-1.95 (0.60)	<0.001		
inf:p.Firmi	0.06 (0.52)	0.90		
inf:p.Bactero	2.03 (0.71)	0.01		

Table 3 The estimated parameters from joint method with (A) MLMM and (B) DMMM. Both models were fitted with Gauss–Hermite quadrature proc NLMIXED in SAS with 5 adaptive quadrature points

(A) Joint model with MLMM

Fixed effects			Random effects		
Parameters	Estimate (s.e)	p-values	Parameters	Estimate (s.e)	p-values
Multivariate			$\sigma_{u_{C_2}}^2$	1.88 (0.35)	<.0001
ξ_{02}^C	-3.46 (0.18)	<.0001	$\sigma_{u_{S_2}}^2$	-0.02 (0.05)	0.75
ξ_{12}^C	0.79 (0.03)	<.0001	$\sigma_{u_{C_3}}^2$	0.31 (0.06)	<.0001
ξ_{03}^C	-0.96 (0.07)	<.0001	$\sigma_{u_{S_3}}^2$	-2.6E-04 (0.02)	0.99
ξ_{13}^C	-0.33 (0.02)	<.0001	$\sigma_{u_y}^2$	0.03 (0.06)	0.56
Continuous			ρ	0.07 (0.13)	0.56
ξ_0^Y	2.19 (0.05)	<.0001	σ_ϵ^2	0.36 (0.03)	<.0001
ξ_1^Y	0.09 (0.07)	0.21			

(B) Joint model with DMMM

Fixed effects			Random effects		
Parameters	Estimate (s.e)	p-values	Parameters	Estimate (s.e)	p-values
Multivariate			$\sigma_{u_{C_1}}^2$	1.81E-04 (0.02)	0.99
λ_2^F	-0.25 (0.21)	0.24	$\sigma_{u_{C_2}}^2$	0.01 (0.04)	0.78
λ_2^E	-2.67 (0.16)	<.0001	$\sigma_{u_{C_3}}^2$	4.37E-08 (1.07E-04)	0.99
λ_3^E	-1.02 (0.09)	<.0001	$\sigma_{u_{S_1}}^2$	4.16E-03 (0.02)	0.83
λ_{22}^{EF}	0.36 (0.21)	0.1	$\sigma_{u_{S_2}}^2$	-7.31E-03 (0.02)	0.74
λ_{32}^{EF}	-0.05 (0.13)	0.72	$\sigma_{u_{S_3}}^2$	-3.03E-08 (6.8E-05)	0.99
Continuous			$\sigma_{u_y}^2$	0.02 (0.04)	0.59
ξ_0^Y	2.19 (0.05)	<.0001	σ_ϵ^2	0.13 (0.02)	<.0001
ξ_1^Y	0.09 (0.07)	0.19	θ	0.15 (0.02)	<.0001

Note: SAS might give negative variances

matrix Σ , where Σ is defined in equation (6). The estimated parameters of the fixed-effects and random-effects parameters are tabulated in Table 3A. Infection has no significant association with the cytokine response. However, it is significantly

associated with the change of ratio of Bacteroidetes: Firmicutes ($\hat{\xi}_{12}^C = 0.79$ with s.e. of 0.03, p -value of $< .0001$) and pooled: Firmicutes ($\hat{\xi}_{13}^C = -0.33$ with s.e. of 0.02, p -value of $< .0001$). We observed that the two outcomes are not correlated, i.e., the estimates of the variances of the random shared effects u_{S_2} and u_{S_3} are almost zero ($\sigma_{u_{S_2}}^2 = 0.002$ with a s.e. of 0.010, p -value of 0.796; $\sigma_{u_{S_3}}^2 = 0.006$ with s.e. of 0.015, p -value of 0.628).

A similar conclusion was obtained from fitting the joint model with the Dirichlet-multinomial mixed model. The estimated parameters are presented in Table 3B. The estimated effect of infection is the same as in the joint method with the multinomial logistics mixed model. The parameter λ_{22}^{EF} represents the log odds ratio of Bacteroidetes to Firmicutes when subjects were infected compared to uninfected. Based on this model, the infection status is not significantly associated with the log odds ratio of Bacteroidetes to Firmicutes.

To further investigate the relationship between the MB composition and the cytokines, we estimated the variance of shared effect in subjects who remained uninfected. A total of 16 subjects were helminth-uninfected at pre-treatment and remained uninfected at 21 months after the first treatment. The estimated parameters are listed in Table 7. We observed that the estimated variance of the shared effect was larger in this subset than in the total sample. When using the Dirichlet-multinomial mixed model for these 16 subjects, the likelihood failed to converge.

The two approaches differ in how the measurement error for the multivariate count outcome is modelled. For the whole dataset and the MLMM, we notice that the variances of random effect $\sigma_{u_{C_2}}^2$ and $\sigma_{u_{C_3}}^2$ are significant. These random effects represent the measurement error as well as the correlation over time. For the DMMM, the measurement error is captured by the overdispersion parameter θ , which is also significant.

The marginal correlations for both observed and those estimated from the joint methods are given in Table 4A–C. In general, the marginal correlation among the categories obtained from the joint method with MLMM is closer to the observed marginal correlation than the correlations obtained from DMMM. The estimated marginal correlation between the categorical and continuous outcomes is different from the observed correlations for both models.

6 Discussion

We proposed two joint models to assess the relationship between a continuous marker and the MB composition while taking into account a set of covariates and measurement error. The methodology was illustrated by a study on the association between helminth infection status, MB composition, and cytokine responses in a longitudinal study in Indonesia. For the MB data, we considered the multinomial logistic mixed model approach [8] and the Dirichlet-multinomial mixed model. To model extra variation due to measurement error or unobserved heterogeneity in the

Table 4 The observed (A) and estimated marginal correlations (B) from the joint method with MLMM and (C) from the joint method with DMMM

(A) The observed marginal correlation from the dataset								
	$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	Y_2
$C_1^{(1)}$	1							
$C_2^{(1)}$	-0.545	1						
$C_3^{(1)}$	-0.53	-0.422	1					
Y_1	0.075	-0.089	0.009	1				
$C_1^{(2)}$	0.262	-0.067	-0.216	0.133	1			
$C_2^{(2)}$	-0.072	-0.07	0.148	-0.029	-0.55	1		
$C_3^{(2)}$	-0.228	0.141	0.104	-0.123	-0.605	-0.331	1	
Y_2	-0.224	0.174	0.067	0.235	0.072	-0.123	0.036	1
(B) The estimated marginal correlation from joint method with MLMM								
	$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	Y_2
$C_1^{(1)}$	1							
$C_2^{(1)}$	-0.58	1						
$C_3^{(1)}$	-0.59	-0.32	1					
Y_1	0.01	-0.02	0.01	1				
$C_1^{(2)}$	0.96	-0.53	-0.60	0.01	1			
$C_2^{(2)}$	-0.56	0.91	-0.26	-0.02	-0.54	1		
$C_3^{(2)}$	-0.57	-0.24	0.91	0.01	-0.63	-0.31	1	
Y_2	0.01	-0.02	0.01	0.12	0.01	-0.02	0.01	1
(C) The estimated marginal correlation from joint method with DMMM								
	$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	Y_2
$C_1^{(1)}$	1							
$C_2^{(1)}$	-0.39	1						
$C_3^{(1)}$	-0.85	-0.15	1					
Y_1	0.02	-0.02	-0.01	1				
$C_1^{(2)}$	0.01	-0.00	-0.01	0.02	1			
$C_2^{(2)}$	-0.00	0.01	0.00	-0.02	-0.38	1		
$C_3^{(2)}$	-0.01	0.00	0.01	-0.01	-0.86	-0.15	1	
Y_2	0.02	-0.02	-0.01	0.14	0.02	-0.02	-0.01	1

multinomial type data, either the conjugate (DMMM) or the normal distribution (MLMM) was used.

We compared our models with a naive approach, which includes bacterial proportions as a covariate in a linear mixed model ignoring the measurement error in the MB data. Our simulation study showed that the estimator of the parameter modelling the effect of the covariate on the continuous outcome in the naive approach was unbiased but less efficient. In our data application, we found

a significant association between the MB and the cytokine response in uninfected subjects using the naive approach. However, this was not confirmed in the joint model approach. Unfortunately, we do not have sufficient information to conclude about this relationship. Our study is underpowered with only 16 subjects who were uninfected and remained uninfected.

The application of the joint model to all subjects showed a significant association between helminth infection on MB composition but not on the cytokine response. Concerning the estimated correlation between the categorical and continuous outcomes, we observed small correlations for both models. For MLMM, the estimated variance of shared effect was $\sigma^2_{us_2} = -0.02$ (s.e. of 0.05) and $\sigma_{us_3}^2$ of -0.000026 (s.e. of 0.02). The measurement errors were relatively large and significant for both the bacterial count outcomes in both approaches. The joint approach with MLMM appeared to represent the marginal Pearson correlations among the categorical counts better. However, the Pearson correlation between the two outcomes was not well represented by both models. The reasons might be that the random effects are not normally distributed or that the used correlation structure is too simple.

To assess model fit, we calculated Pearson's correlation coefficients from the observed data and compared these with the marginal correlations given by the model. For the data example, the MLMM model appeared to represent the observed correlations among the categorical counts better, and therefore, we may conclude that the MLMM fits the data better than the DMMM. The development of formal goodness-of-fit tests to decide among the various models is a topic of future research. Finally, such a test might be based on non-parametric measures for the correlation since the counts are non-normal.

We proposed here the joint model between the multivariate count of three categories and continuous outcome. It appeared to be challenging to fit these models. For instance, the optimization under the approximation of Gauss–Hermite (the so-called adaptive Gauss–Hermite) in R did not always converge; especially when the variance of random effect is small. To overcome this, we opted to use SAS, which has a built-in procedure called proc NLMIXED, designed for mixed-effect models for the data analysis. Some of the estimates for the variances appeared to be negative, which suggests that the convergence problems in R might be caused by restrictions on the parameter space. A limitation of our model is the small number of categories which it can handle. It would be attractive to extend the model to more than three categories. However, the computational burden will increase further. With an increasing number of categories (high-dimensional), a penalty function might be needed to deal with a large number of variance components [2]. These are topics for future research.

In this chapter, the focus is on modelling of the relationships between two outcome variables and a set of covariates. When the interest is only on testing for associations, fitting these models might be too time-consuming. A score test needs to be derived for testing the null hypothesis of no association between the compositional and continuous outcomes. Another approach is to use the compositional parameter as a covariate in the model and test for association using a Wald test. However, the power might be small in the presence of a large measurement error.

More research is needed to derive test statistics and to study their performance under a range of scenarios.

Joint models better reflect the underlying biological mechanisms since they enable the formulation of unmeasured mechanisms by introducing random (latent) factors for modelling the relationship between variables. The next step is to embed these models in a causal inference framework. In such a framework, the paths among all variables can be drawn, and confounding factors like BMI or diet can be included. The study presented in this chapter is a subset of a larger randomized clinical trial. Subjects were randomized to receive antihelminthic treatment or placebo. Under some conditions, this randomization might be used to perform causal inference about the relationship among the infection status, the MB composition, and the cytokine response in the subset of this chapter [14, Chapter 6].

To conclude, although the joint models are challenging to fit when the outcomes are from different types, they might give more insight into three-way relationships between a covariate and two outcomes.

7 Software

Two statistical software was used in this chapter: R in the simulation study and SAS for model fitting. SAS has a built-in function `proc NL MIXED` to compute a likelihood containing integration of random effect, especially with complex covariance structure as in our proposed methods. Hence, computational time in SAS is faster than that in R. R codes and the dataset used in this chapter can be found in <https://github.com/IvonneMartin/JMoverCat>.

R code to generate datasets

The code for generating dataset here is applied to a dataset with one binary covariate and the multivariate count outcome consisting of three categories with total count per sample is 2000. The dataset used in the computation assumes a wide format as given in Table 5.

Here, the superscripts represent the time point where the observation takes place.

Table 5 An example data of wide format for computation in R

ID	$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	$X^{(1)}$	$X^{(2)}$	$Y^{(1)}$	$Y^{(2)}$
1	1538	36	426	1364	36	600	1	1	2.53	2.23
2	1306	13	681	1098	584	318	1	1	2.25	2.59
:	:	:	:	:	:	:	:	:	:	:
N	1408	52	540	1193	41	766	1	1	2.58	1.38

The function `DataGenerate` (lines 30–127) is the core function to generate the dataset, which will call the subroutine `FixEf` (line 6–19) to define the linear model. The procedure is described at the end of this section.

```

1 ## FixEf returns a multiplication between a design
2 ## matrix
3 ## (Des) with a vector of regression coefficients (b).
4 ## x.vec : the value of binary covariate.
5 ## method : "dirmult" or "multl".
6
7 FixEf <- function(x.vec,b,Des,method ){
8   gt <- c(Des %*% b)
9
10  if (method == "dirmult"){
11    if (x.vec == 0) {
12      g = gt[seq(1,nrow(Des),by=2)]
13    } else {
14      g = gt[seq(2,nrow(Des),by=2)] }
15  if(method == "multl") {
16    if (x.vec == 0) {
17      g = gt[seq(1,nrow(Des),by=2)]
18    } else {
19      g = gt[seq(2,nrow(Des),by=2)] }
20
21
22 ## DataGenerate is the core function to generate
23 ## datasets.
24 ## N : the number of subjects
25 ## method : the model for the multivariate count
26 ## outcome
27 ## ("dirmult" or "multl")
28 ## var.level : variance level ("high" or "low", as
29 ## defined in
30 ## the simulation setting).
31
32
33
34
35 Q <- 3      # number of categories
36 S <- 2000    # total count
37 uY <- 0.9    # std. dev. for continuous outcome

```

```

38   eps <- 0.7      # std. dev. for residuals in
39           # continuous outcome
40
41   if (method == "dirmult") {
42     th <- 0.1
43     uC <- c(1,1,0.9)
44     if(var.level == "high") {
45       uS <- c(1,1,0.9) }
46     if (var.level == "low") {
47       uS <- c(1,0.5,0.6) }
48
49   Sigma1 <- matrix(0,nrow = 4,ncol = 4)
50   diag(Sigma1) <- c(uC^2 + uS^2, sum(uS^2) + uY^2)
51   Sigma1[4,1] <- uS[1]^2
52   Sigma1[4,2] <- uS[2]^2
53   Sigma1[4,3] <- uS[3]^2
54   Sigma <- as.matrix(Matrix:::forceSymmetric(Sigma1,
55   uplo = "L"))
56
57   Des <- as.matrix(cbind(rep(1,6),rep(0:1,3),c
58     (0,0,1,1,0,0),
59     c(rep(0,4),rep(1,2)),c(0,0,0,1,0,0),c(rep(0,5),1)))
60   beta <- c(0,0.5,-3.5,-1.3,0.8,-0.15,-2,1)
61   Umc <- rmvnorm(N,mean=rep(0,4),sigma=Sigma)
62   Eps <- rnorm(2*N, mean = 0, sd = eps)
63   X <- cbind(rbinom(N,size=1,prob = 0.5),rbinom(N,
64     size = 1,prob = 0.5))
65
66   XB1 <- t(sapply(X[,1],FixEf,b = beta[1:6],Des,
67     method))
68   XB.tilde1 <- (1/th)*exp(cbind(XB1+Umc[,1:3]))
69   C1 <- t(apply(XB.tilde1,1,Dirichlet.multinomial,Nrs
70     = S))
71   Y1 <- cbind(1,X[,1])%*%beta[7:8] + Umc[,4] + Eps[1:
72     N]
73
74   XB2 <- t(sapply(X[,2],FixEf,b = beta[1:6],Des,
75     method))
76   XB.tilde2 <- (1/th)*exp(cbind(XB2+Umc[,1:3]))
77   C2 <- t(apply(XB.tilde2,1,Dirichlet.multinomial,Nrs
78     = S))
79   Y2 <- cbind(1,X[,2])%*%beta[7:8] + Umc[,4] + Eps[(N
80     +1):(2*N)]

```

```

74
75     data1 <- cbind(C1,C2,X,Y1,Y2)
76 }
77
78 if(method == "mult1") {
79
80     rho <- 0.1
81     uC <- c(1,0.8)
82     if(var.level == "high") {
83         uS <- c(1,0.9)
84     }
85     if (var.level == "low") {
86         uS <- c(0.5,0.6)
87     }
88
89     Sigma1 <- matrix(0,nrow = 3,ncol = 3)
90     diag(Sigma1) <- c(uC^2 + uS^2, sum(uS^2) + uY^2)
91     Sigma1[2,1] <- rho*uC[1]*uC[2]
92     Sigma1[3,1] <- uS[1]^2
93     Sigma1[3,2] <- uS[2]^2
94     Sigma <- as.matrix(Matrix::forceSymmetric(Sigma1,
95                           uplo = "L"))
96
97     Des <- cbind(c(rep(1,2),rep(0,2)),c(rep(0,2),rep
98                  (1,2)),
99                  c(0,1,0,0),c(0,0,0,1))
100    beta <- c(-3.5,-1.3,0.8,-0.15,-2,0.1)
101
102
103    eta1 <- t(sapply(X[,1],FixEf,b = beta[1:4],Des,
104                  method))
105    eta2 <- t(sapply(X[,2],FixEf,b = beta[1:4],Des,
106                  method))
107
108    Eta1 <- cbind(1, exp(eta1 + Umc[,c(1,2)]))
109    SumEta1 <- apply(Eta1,1,sum)
110    Eta1 <- Eta1/SumEta1
111
112    Eta2 <- cbind(1, exp(eta2 + Umc[,c(1:2)]))
113    SumEta2 <- apply(Eta2,1,sum)
114    Eta2 <- Eta2/SumEta2
115

```

```

114 # Generate the outcomes
115 C1 <- t(apply(Eta1,1,rmultinom,n=1,size = S))
116 C2 <- t(apply(Eta2,1,rmultinom,n=1,size = S))
117
118 Y1 <- cbind(1,X[,1])%*%beta[5:6] + Umc[,3] + Eps[1:N]
119 Y2 <- cbind(1,X[,2])%*%beta[5:6] + Umc[,3] +
120 Eps[(N+1):(2*N)]
121
122 data1 <- cbind(C1,C2,X[,1],X[,2],Y1,Y2)
123 }
124 colnames(data1) <- c("C11","C21","C31","C12","C22",
125 "C32",
126 "X1","X2","Y1","Y2")
127 return(data1)
128 }
129 # examples
130 #DataGenerate(N = 50,method = "dirmult",var.level =
high)

```

For each method for the multivariate count outcome (`dirmult` or `multl`), we first specify standard deviations for random effects to formulate the covariance matrix `Sigma` (lines 42–55 for `dirmult` and lines 80–94 for `multl`). The design matrix for the covariate `Des` is to obtain the linear regressor for all logits. `beta` is the vector consisting of all fixed-effect parameters as defined in the simulation setting. After generating random effects (`Umc`) following the covariance structure, we generate a model matrix for a multivariate outcome (`Eta1` and `Eta2`, for the first and second time points, respectively).

R code for the computation of the likelihood of joint model

The function `loglik` (lines 253–321) is the core function to compute the log-likelihood of the joint method both with Dirichlet-multinomial mixed model or multinomial logistics mixed model. The function requires four subroutines, namely the function `CNF` to compute the log-likelihood of the multivariate count outcome, the function `F` to compute the total likelihood, `integralComputation` to compute the Gauss–Hermite approximation to the log-likelihood, and `mgauss.hermite` function to call for the nodes and weights of Gauss–Hermite polynomial. The procedure is described briefly at the end of this section.

```

131 ## CNF returns the marginal log-likelihood for the
multivariate

```

```

132 ## count outcome.
133 ## Ct      : the count vector
134 ## g       : the output from FixEf function
135
136 CNF <- function(Ct,g,method) {
137   gs <- sum(g)
138   ys <- sum(Ct)
139
140   if (method == "dirmult") {
141     val <- lgamma(ys+1) + lgamma(gs) - lgamma(ys+gs) +
142           sum(lgamma(Ct+g) - lgamma(g) - lgamma(Ct+1))
143   }
144
145   if (method == "multl") {
146     prob <- g/gs
147     val <- lgamma(ys + 1) + sum(Ct * log(prob) -
148                               lgamma(Ct + 1))
149   }
150   return(val)
151
152
153 ## F returns the log-likelihood value for a subject for
154 ## a given
155 ## Gauss-Hermite node z.
156 ## b the fixed effect parameter values
157 ## theta : the overdispersion parameter for dirmult.
158 ## Sigma : the covariance matrix for the random effect.
159 ## ev    : the standard deviation for residuals in
160 ##         continuous
161 ##         outcome.
162 ## data   : the observations for a subject.
163
164
165
166   if (method == "dirmult") {
167     eta1 <- FixEf(Yt[7], b[1:6],Des,method)
168     eta2 <- FixEf(Yt[8], b[1:6],Des,method)
169     eta1 <- (1/theta)*exp(eta1 + c(z[1:3]))
170     eta2 <- (1/theta)*exp(eta2 + c(z[1:3]))
171     val <- CNF(Yt[1:3],eta1,method) + CNF(Yt[4:6],
172         eta2,method) +

```

```

172      dnorm(Yt[9],mean=c(1,Yt[7])%*%b[7:8] + z[4],sd
173          = ev,log=TRUE) +
174      dnorm(Yt[10],mean=c(1,Yt[8])%*%b[7:8] + z[4],sd
175          = ev,log=TRUE) +
176      dmvnorm(z, mean = rep(0,4), sigma = Sigma, log=
177          TRUE)
178      return(val)
179  }
180
181  if (method == "mult1") {
182      eta1 <- FixEf(Yt[7], b[1:4],Des,method)
183      eta2 <- FixEf(Yt[8], b[1:4],Des,method)
184      eta1 <- exp(eta1 + z[1:2])
185      eta2 <- exp(eta2 + z[1:2])
186      Eta1 <- c(1,eta1)
187      Eta2 <- c(1,eta2)
188      val <- CNF(Yt[1:3],Eta1,method) + CNF(Yt[4:6],
189          Eta2,method) +
190      dnorm(Yt[9],mean=c(1,Yt[7])%*%b[5:6] + z[3],sd
191          = ev,log=TRUE) +
192      dnorm(Yt[10],mean=c(1,Yt[8])%*%b[5:6] + z[3],sd
193          = ev,log=TRUE) +
194      dmvnorm(z, mean = rep(0,3), sigma = Sigma, log=
195          TRUE)
196  }
197  return(val)
198 }
199
200
201 ## integralComputation returns the likelihood
202 ## approximation
203 ## for subject i based on adaptive
204 ## Gauss-Hermite quadratures.
205 ## b : fixed effect parameters
206 ## theta : overdispersion parameter for dirmult
207 ## ev : standard deviation for residuals in
208 ##       continuous outcome.
209 ## x, w : nodes and weights from multivariate
210 ##       Gauss-Hermite quadrature.
211
212
213 ## integralComputation<-function(i,b,theta,Sigma,ev,
214 ## dataset,
215 ##           x,w,f,method) {
216 data <- dataset[i,]

```

```

207 ff <- function(z) f(z,b,theta,Sigma,ev,data,method)
208
209 if (method == "dirmult"){
210 opt <- try(optim(c(0.1,0.1,-0.2,0.1),ff,method="BFGS
211   ",
212   control=list(fnscale = -1,maxit=9000),hessian=TRUE) )
213 FIM <- nearPD(-opt$hessian)$mat
214 invDer <- solve(FIM)
215 CH <- chol(invDer)
216 a.star <- t(opt$par + (sqrt(2)*CH%*%t(x)))
217 y <- apply(a.star,1,ff)
218 L <- (2^(4/2))*sqrt(det(CH))*sum(exp(y)*w*exp(x^2))
219
220 if (method == "multl"){
221   opt <- try(optim(c(0.1,-0.2,0.1),ff,method="BFGS",
222     control=list(fnscale = -1,maxit=9000),hessian=TRUE)
223   )
224   FIM <- nearPD(-opt$hessian)$mat
225   invDer <- solve(FIM)
226   CH <- chol(invDer)
227   a.star <- t(opt$par + (sqrt(2)*CH%*%t(x)))
228   y <- apply(a.star,1,ff)
229   L <- (2^(3/2))*sqrt(det(CH))*sum(exp(y)*w*exp(x^2))
230
231 return(L)
232 }
233
234
235
236
237
238 ## mgauss.hermite returns a set of nodes and weights
239 ## from
240 ## multivariate Gauss - Hermite quadrature.
241 ## n      : the number of quadratures
242 ## mu     : mean vector
243 ## Sigma  : covariance matrix
244
245 mgauss.hermite <- function(n, mu, sigma) {
246   if(!all(dim(sigma) == length(mu)))
247     stop("mu and sigma have nonconformable dimensions")
248
249   dm <- length(mu)
250   gh <- gauss.hermite(n)
251   idx <- as.matrix(expand.grid(rep(list(1:n),dm)))
252   pts <- matrix(gh[idx,1],nrow(idx),dm)
253   wts <- apply(matrix(gh[idx,2],nrow(idx),dm), 1, prod)

```

```
247
248     return(list(points=pts, weights=wts))
249 }
250
251
252 ## loglik is the core function to calculates the total
253 ## likelihood
254 ## given the dataset.
255 ## params should be in the following format:
256 ## params = {betas_C, betas_Y,uC's, uS's,theta or rho,
257 ## uY,ev}
258
259 loglik <- function(params,method,data) {
260   require(ecoreg)
261   require(mvtnorm)
262   require(Matrix)
263
264   var.Y <- exp(params[(length(params)-1):length(params)
265                 ])
266   uY <- var.Y[1]
267   ev <- var.Y[2]
268   if (method == "dirmult") {
269     var.Sigma <- exp(params[(length(params) - 8):
270                         (length(params) - 2)])
271     uC <- var.Sigma[1:3]
272     uS <- var.Sigma[4:6]
273     theta <- var.Sigma[7]
274     beta <- c(0,params[1:(length(params) - 9)])
275
276     Sigma1 <- matrix(0,nrow=4,ncol=4)
277     diag(Sigma1) <- c(uC^2 + uS^2,sum(uS^2) + uY^2)
278     Sigma1[4,1] <- uS[1]^2
279     Sigma1[4,2] <- uS[2]^2
280     Sigma1[4,3] <- uS[3]^2
281     Sigma <- as.matrix(Matrix:::forceSymmetric(
282       Sigma1,
283       uplo = "L"))
284
285     pts <- mgauss.hermite(3, mu=rep(0,3), sigma=
286                           Sigma)
287     xGH <- pts$points
288     wGH <- pts$weights
289
290     index<-seq(1,nrow(data),by=1)
```

```

286      w <- wGH
287      x <- as.matrix(xGH )
288      f <- match.fun(F)
289      e1 <- sapply(index,integralComputation,b = beta
290                  ,
291                  theta = theta, Sigma = Sigma,ev,dataset = data
292                  ,x = x,
293                  w = w,f = f,method = method)
294
295      res <- sum(log(e1))
296      return(res)
297  }
298
299  if (method == "multl"){
300    rho <- cos(params[length(params) - 2])
301    var.Sigma <- exp(params[(length(params) - 6):
302                        (length(params) - 3)])
303    uC <- var.Sigma[1:2]
304    uS <- var.Sigma[3:4]
305    beta <- params[1:(length(params) - 7)]
306
307    Sigma1 <- matrix(0,nrow=3,ncol=3)
308    diag(Sigma1) <- c(uC^2 + uS^2,sum(uS^2) + uY^2)
309    Sigma1[2,1] <- rho*uC[1]*uC[2]
310    Sigma1[3,1] <- uS[1]^2
311    Sigma1[3,2] <- uS[2]^2
312    Sigma <- as.matrix(Matrix::forceSymmetric(Sigma1,
313                                         uplo = "L"))
314
315    pts <- mgauss.hermite(3, mu=rep(0,3), sigma=Sigma
316                           )
317
318    index<-seq(1,nrow(data),by=1)
319    w <- wGH
320    x <- as.matrix(xGH )
321    f <- match.fun(F)
322    e1 <- sapply(index,integralComputation,b = beta,
323                  theta = 1, Sigma = Sigma,ev,dataset = dat1,
324                  x = x,w = w,f = f,method = method)
325
326    res <- sum(log(e1))

```

```
327     }
328 return(res)
```

The function `loglik` requires an input of initial values of parameters, both the fixed effect and standard deviations of the covariance structure (`params`), a method for the multivariate count outcome (`method`, either `dirmult` or `mult1`), and a dataset in the wide format as defined in Table 5.

The function first specifies the parameters `params` into fixed-effect parameters and standard deviations of random effects. Depending on the method for the multivariate count outcome, the function specifies the covariance structure for the random effect `Sigma`.

Using the covariance matrix `Sigma`, the subroutine `mgauss.hermite` is used to call the nodes and weights from the standard Gauss–Hermite polynomial. The subroutine `integralComputation` is used to compute the likelihood contribution for each subject.

SAS code: joint method with multinomial logistics mixed model

For computation in SAS, the dataset is in the long format as given in Table 6.

Often, the likelihood computation gives a singular Hessian matrix. In this case, the following Cholesky factorization of the covariance matrix is used:

$$\Sigma_3 = \begin{pmatrix} \tau_1 & 0 & 0 \\ \tau_{12} & \tau_2 & 0 \\ \tau_{13} & \tau_{23} & \tau_3 \end{pmatrix} \begin{pmatrix} \tau_1 & \tau_{12} & \tau_{13} \\ 0 & \tau_2 & \tau_{23} \\ 0 & 0 & \tau_3 \end{pmatrix}. \quad (12)$$

For each SAS code, the `llY` and `llC` are the log-likelihood for continuous and multivariate outcomes, respectively. Note that there is a variable `yt` in the dataset consisting of 1's for all samples. This is a dummy outcome variable.

```
proc nlmixed data = bact2time qpoints=5;
/*fixed effect parameters, th's are for the intercepts
and b_if's for the infection effect*/
```

Table 6 The dataset in the long format for computation in SAS

id	t1	t2	t3	inf	tpoint	respY	yt
1	1538	36	426	1	0	2.53	1
1	1364	36	600	1	1	2.23	1
2	1306	13	681	1	0	2.25	1
2	1098	584	318	1	1	2.59	1
:	:	:	:	:	:	:	:
N	1408	52	540	1	0	2.58	1
N	1193	41	766	1	1	1.38	1

```

parms thC2 = 1 b_ifC2 = 0.3
      thC3 = 2 b_ifC3 = -0.2
      thY = 0.5 b_ifY = 1
      logse = -0.9
/* The parameters for Cholesky factorization of Sigma*/
      tau1 = -1.5 tau12 = -0.4 tau2 = 0.5 tau13 = -2
      tau23 = -1.5
      tau3 = 1.5;

/* eta's are the linear model each category. */
      eta1 = 0;
      eta2 = thC2 + b_ifC2*inf + b1;
      eta3 = thC3 + b_ifC3*inf + b2;

array exp_eta{3};
exp_eta1 = 1;
exp_eta2 = exp(eta2);
exp_eta3 = exp(eta3);
bot = exp_eta1+exp_eta2+exp_eta3;

/*the linear model for the continuous outcome*/
mY = thY + b_ifY*inf + b3;
se = exp(logse);

lly = -0.5*log(2*3.1415) - log(se) -
0.5*((respY - mY)**2)/se**2;

llC = t1*log(exp_eta1/bot) + t2*log(exp_eta2/bot) +
t3*log(exp_eta3/bot);

model yt ~ GENERAL(llC + lly);
s11 = tau1*tau1;
s21 = tau1*tau12;
s22 = tau12*tau12 + tau2*tau2;
s31 = tau1*tau13;
s32 = tau12*tau13 + tau2*tau23;
s33 = tau13*tau13 + tau23*tau23 + tau3*tau3;
random b1 b2 b3 ~ NORMAL([0,0,0],[s11,s21,s22,s31,s32,s33])
subject=id;

estimate 'variance of u2' tau1*tau1 - tau1*tau13;
estimate 'variance of us2' tau1*tau13;
estimate 'variance of u3' tau12*tau12 + tau2*tau2 -
tau12*tau13 + tau2*tau23;
estimate 'variance of us3' tau12*tau13 + tau2*tau23;
estimate 'variance of uy' tau13*tau13 + tau23*tau23 +
tau3*tau3 - tau1*tau13 - tau12*tau13 - tau2*tau23;
estimate 'rho' (tau1*tau12)/sqrt(tau1*tau1 - tau1*tau13)*sqrt(

```

```
tau12*tau12 + tau2*tau2 - (tau12*tau13 + tau2*tau23));
run;
```

SAS code for joint model with Dirichlet-multinomial regression

```
proc nlmixed data = bact2time qpoints= 5;
/*fixed effect parameters. Here b_2B, b_2A,...,b_32AB
are lambdas in the simulation setting.*/
parms b2B = -0.10 b2A = -2 b3A= -1.2 b22AB = 0.3 b32AB= -0.2
      thY = 0.5 b_ifY = 1
      logse = -0.9
      logth = -2.46924285
      tau11 = 0.5 tau21= -0.5 tau22= 1.2 tau31=-1.2 tau32=3.2
      tau33=2.1 tau41=-2.3 tau42=2.1 tau43=1.2 tau44=-1.5;

theta = exp(logth); /*overdispersion parameter */
se = exp(logse);

if (inf = 0) then
  do;
    eta1 = (1/theta)*exp(0 + b1);
    eta2 = (1/theta)*exp(0 + b2A + b2);
    eta3 = (1/theta)*exp(0 + b3A + b3);
  end;
else
  do;
    eta1 = (1/theta)*exp(0 + b2B + b1);
    eta2 = (1/theta)*exp(0 + b2B + b2A + b22AB + b2);
    eta3 = (1/theta)*exp(0 + b2B + b3A + b32AB + b3);
  end;

gs = eta1 + eta2 + eta3;
ys = 2000;

llC = lgamma(ys+1) + lgamma(gs) - lgamma(ys+gs) +
(lgamma(t1+eta1) + lgamma (t2+eta2) + lgamma(t3+eta3) -
lgamma(eta1) - lgamma(eta2) - lgamma(eta3) -
lgamma(t1+1) - lgamma(t2+1) - lgamma(t3+1));

mY = thY + b_ifY*inf + b4;
lly = -0.5*log(2*3.1415) - log(se)
- 0.5*((respY - mY)**2)/se**2;
```

```

model yt ~ GENERAL(l1C + l1Y);
s11 = tau11*tau11;
s21 = tau11*tau21;
s22 = tau21*tau21 + tau22*tau22;
s31 = tau11*tau31;
s32 = tau21*tau31 + tau22*tau32;
s33 = tau31*tau31 + tau32*tau32 + tau33*tau33;
s41 = tau11*tau41;
s42 = tau21*tau41 + tau22*tau42;
s43 = tau31*tau41 + tau32*tau42+tau33*tau43;
s44 = tau41*tau41 + tau42*tau42 + tau43*tau43
    + tau44*tau44;
random b1 b2 b3 b4 ~ NORMAL([0,0,0,0],[s11,s21,s22,s31,s32,
s33,s41,s42,s43,s44]) subject=id;

estimate 'var(uS1)' tau11*tau41;
estimate 'var(uS2)' tau21*tau41 + tau22*tau42;
estimate 'var(uS3)' tau31*tau41 + tau32*tau42 + tau33*tau43;
estimate 'var(uC1)' tau11*tau11 - tau11*tau41;
estimate 'var(uC2)' tau21*tau21 +
tau22*tau22 - (tau21*tau41 + tau22*tau42);
estimate 'var(uC3)' tau31*tau31 + tau32*tau32 + tau33*tau33 -
(tau31*tau41 + tau32*tau42 + tau33*tau43);
estimate 'var(uY)' tau41*tau41 + tau42*tau42 + tau43*tau43 +
tau44*tau44 - (tau11*tau41) - (tau21*tau41 + tau22*tau42) -
(tau31*tau41+tau32*tau42+tau33*tau43);
estimate 'theta' exp(logth);
run;

```

Acknowledgments The research leading to these results has received funding from the European Union’s Horizon 2020 programme H2020-MSCA-ITN under grant agreement 721815 (IMforFUTURE), from The Royal Netherlands Academy of Arts and Science (KNAW), Contract: 57-SPIN3-JRP, and from the Directorate General of Resources for Science Technology and Higher Education (DGRSTHE) of Indonesia—Leiden University.

Appendix

See Figs. 5, 6 and Tables 7, 8, 9, 10, 11.

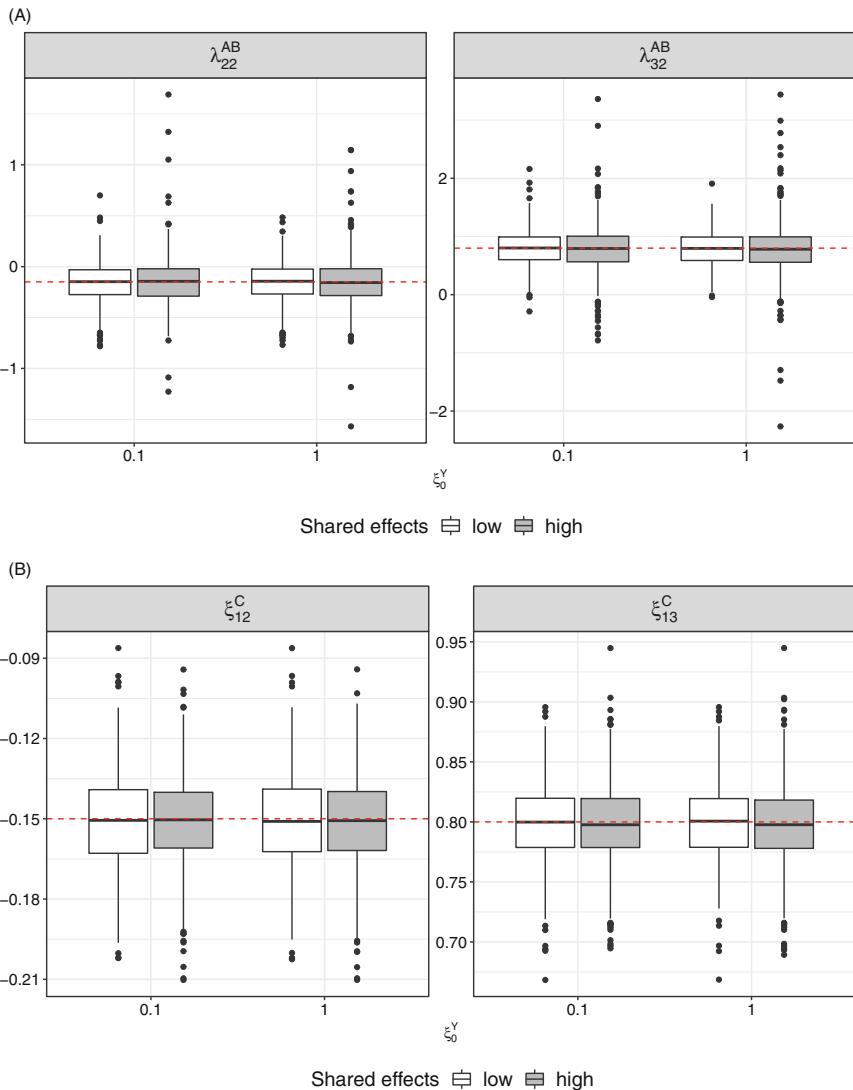


Fig. 5 Simulation results: the estimated covariates of the bacterial categories from joint method with DMMM (Panel a) and joint method with MLMM (Panel b). The red dashed lines represent the true parameter

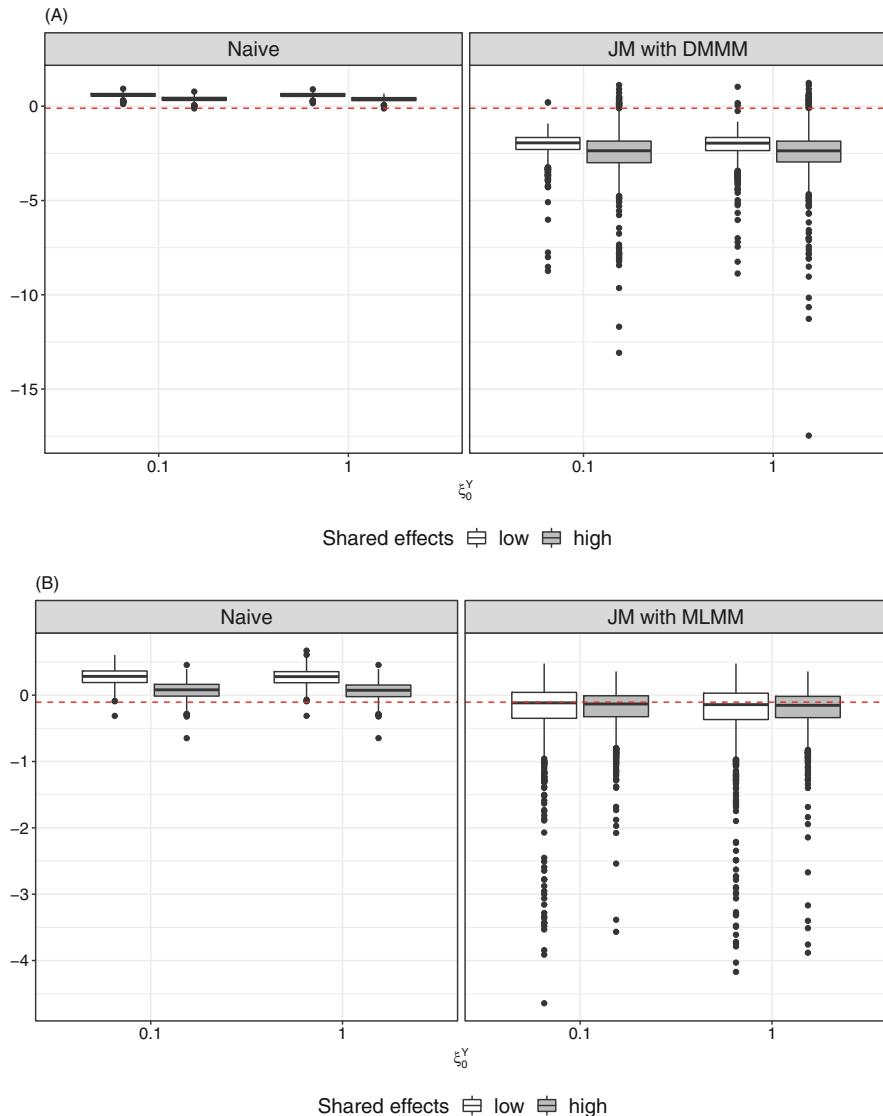


Fig. 6 The estimated variability of the continuous outcome ($\log(\sigma_{u_t})$) from joint method with DMMM (a) and MLMM (b)

Table 7 Data analysis: the joint model in the longitudinal setting in subjects who were helminth-uninfected at pre-treatment and remained uninfected at 21 months after the first treatment ($N = 16$). The model fitting used SAS with 10 quadrature points

Fixed effects	Estimate (95%CI)	p-value
Intercepts		
ξ_1^Y	2.12 (1.93, 2.31)	< .0001
ξ_{02}^C	-3.02 (-3.65, -2.38)	< .0001
ξ_{03}^C	-1.01 (-1.26, -0.77)	< .0001
Random effects	Estimate (s.e)	p-value
$\sigma_{u_{C_2}}^2$	1.499 (0.544)	0.016
$\sigma_{u_{C_3}}^2$	0.204 (0.082)	0.028
$\sigma_{u_{S_2}}^2$	-0.140 (0.107)	0.216
$\sigma_{u_{S_3}}^2$	-0.0004 (0.039)	0.992
$\sigma_{u_Y}^2$	0.156 (0.143)	0.294
σ_ϵ^2	0.208 (0.052)	0.002
ρ	0.314 (0.207)	0.207

Table 8 The observed (A) and estimated (B) marginal correlations from joint model with Dirichlet-multinomial mixed model when the variances of shared effects are low

Joint model with DMM						
Observed marginal correlation	$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$
	1				$C_3^{(2)}$	$C_3^{(2)}$
$C_1^{(1)}$	1					
$C_2^{(1)}$	-0.41 (0.047)	1				
$C_3^{(1)}$	-0.9 (0.02)	-0.03 (0.03)	1			
Y_1	0.19 (0.04)	-0.07 (0.04)	-0.17 (0.04)	1		
$C_1^{(2)}$	0.85 (0.02)	-0.32 (0.04)	-0.78 (0.02)	0.19 (0.04)	1	
$C_2^{(2)}$	-0.32 (0.04)	0.62 (0.06)	0.05 (0.04)	-0.12 (0.04)	-0.41 (0.03)	1
$C_3^{(2)}$	-0.78 (0.02)	0.05 (0.04)	0.83 (0.02)	-0.16 (0.04)	-0.9 (0.02)	-0.03 (0.03)
Y_2	0.19 (0.04)	-0.12 (0.04)	-0.16 (0.04)	0.77 (0.02)	0.19 (0.04)	-0.07 (0.04)
Estimated marginal correlation	$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$
	1				$C_3^{(2)}$	$C_3^{(2)}$
$C_1^{(1)}$	1					
$C_2^{(1)}$	-0.41	1				
$C_3^{(1)}$	-0.90	-0.03	1			
Y_1	0.21	-0.14	-0.17	1		
$C_1^{(2)}$	0.85	-0.32	-0.78	0.21	1	
$C_2^{(2)}$	-0.32	0.61	0.06	-0.14	-0.41	1
$C_3^{(2)}$	-0.78	0.06	0.83	-0.17	-0.90	-0.03
Y_2	0.21	-0.14	-0.17	0.84	0.21	-0.14

Table 9 The observed (A) and estimated (B) marginal correlations from the joint model with Dirichlet-multinomial mixed model when the variances of shared effect are high

(A) Observed marginal correlation		$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	Y_2
$C_1^{(1)}$	1								
$C_2^{(1)}$	-0.41 (0.03)	1							
$C_3^{(1)}$	-0.87 (0.02)	-0.09 (0.03)	1						
Y_1	0.03 (0.05)	0.05 (0.05)	-0.06 (0.05)	1					
$C_1^{(2)}$	0.87 (0.02)	-0.33 (0.04)	-0.77 (0.03)	0.03 (0.04)	1				
$C_2^{(2)}$	-0.33 (0.04)	0.72 (0.05)	-0.03 (0.04)	0.01 (0.04)	-0.41 (0.03)	1			
$C_3^{(2)}$	-0.77 (0.02)	-0.03 (0.04)	0.85 (0.02)	-0.04 (0.04)	-0.87 (0.02)	-0.09 (0.03)	1		
Y_2	0.03 (0.04)	0.01 (0.05)	-0.04 (0.04)	0.83 (0.01)	0.03 (0.04)	0.05 (0.05)	-0.06 (0.04)	1	
(B) Stimulated marginal correlation		$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	Y_2
$C_1^{(1)}$	1								
$C_2^{(1)}$	-0.41	1							
$C_3^{(1)}$	-0.87	-0.09	1						
Y_1	0.04	-0.00	-0.04	1					
$C_1^{(2)}$	0.87	-0.32	-0.77	0.04	1				
$C_2^{(2)}$	-0.32	0.71	-0.02	-0.00	-0.41	1			
$C_3^{(2)}$	-0.77	-0.02	0.85	-0.04	-0.87	-0.09	1		
Y_2	0.04	-0.00	-0.04	0.89	0.04	-0.00	-0.05	1	

Table 10 The observed (A) and estimated (B) marginal correlations from the joint model with multinomial logistics mixed model when the variances of shared effects are low

(A) Observed marginal correlation		$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	Y_2
$C_1^{(1)}$	1								
$C_2^{(1)}$	-0.22 (0.05)	1							
$C_3^{(1)}$	-0.9 (0.02)	-0.21 (0.03)	1						
Y_1	-0.28 (0.04)	0.1 (0.04)	0.23 (0.04)	1					
$C_1^{(2)}$	0.98 (0.00)	-0.19 (0.05)	-0.9 (0.02)	-0.27 (0.04)	1				
$C_2^{(2)}$	-0.19 (0.05)	0.84 (0.02)	-0.18 (0.03)	0.09 (0.04)	-0.22 (0.05)	1			
$C_3^{(2)}$	-0.9 (0.02)	-0.18 (0.03)	0.98 (0.00)	0.24 (0.04)	-0.9 (0.02)	-0.21 (0.03)	1		
Y_2	-0.27 (0.04)	0.09 (0.05)	0.24 (0.04)	0.74 (0.02)	-0.28 (0.04)	0.1 (0.05)	0.23 (0.04)	1	
(B) Estimated marginal correlation		$C_1^{(1)}$	$C_2^{(1)}$	$C_3^{(1)}$	Y_1	$C_1^{(2)}$	$C_2^{(2)}$	$C_3^{(2)}$	Y_2
$C_1^{(1)}$	1								
$C_2^{(1)}$	-0.23	1							
$C_3^{(1)}$	-0.9	-0.21	1						
Y_1	-0.22	-0.1	0.26	1					
$C_1^{(2)}$	0.98	-0.2	-0.9	-0.23	1				
$C_2^{(2)}$	-0.2	0.85	-0.18	0.06	-0.23	1			
$C_3^{(2)}$	-0.9	-0.18	0.98	0.2	-0.9	-0.21	1		
Y_2	-0.23	0.06	0.2	0.49	-0.22	-0.1	0.26	1	

Table 11 The observed (A) and estimated (B) marginal correlations from the joint model with multinomial logistics mixed model when the variances of shared effects are high

(A) Observed marginal correlation								
	$C_1^{(1)}$		$C_2^{(1)}$		$C_3^{(1)}$		Y_1	
$C_1^{(1)}$	1							
$C_2^{(1)}$	-0.29 (0.05)	1						
$C_3^{(1)}$	-0.86 (0.02)	-0.24 (0.03)	1					
Y_1	-0.45 (0.04)	0.26 (0.04)	0.32 (0.04)	1				
$C_1^{(2)}$	0.98 (0.0)	-0.26 (0.05)	-0.86 (0.02)	-0.45 (0.04)	1			
$C_2^{(2)}$	-0.26 (0.05)	0.89 (0.02)	-0.21 (0.03)	0.25 (0.04)	-0.29 (0.05)	1		
$C_3^{(2)}$	-0.86 (0.02)	-0.21 (0.03)	0.99 (0.00)	0.32 (0.04)	-0.86 (0.02)	-0.24 (0.03)	1	
Y_2	-0.45 (0.04)	0.25 (0.04)	0.32 (0.04)	0.84 (0.01)	-0.45 (0.04)	0.26 (0.04)	0.32 (0.04)	1
(B) Estimated marginal correlation								
	$C_1^{(1)}$		$C_2^{(1)}$		$C_3^{(1)}$		Y_1	
$C_1^{(1)}$	1							
$C_2^{(1)}$	-0.3	1						
$C_3^{(1)}$	-0.86	-0.23	1					
Y_1	-0.38	0.1	0.34	1				
$C_1^{(2)}$	0.98	-0.26	-0.86	-0.4	1			
$C_2^{(2)}$	-0.26	0.89	-0.21	0.21	-0.3	1		
$C_3^{(2)}$	-0.86	-0.21	0.99	0.29	-0.86	-0.23	1	
Y_2	-0.4	0.21	0.29	0.64	-0.38	0.1	0.34	1

References

1. Agresti, A.: Categorical Data Analysis, volume 792 of Wiley Series in Probability and Statistics, 3rd edn. Wiley, Hoboken, NJ (2013)
2. Bondell, H.D., Krishna, A., Ghosh, S.K.: Joint variable selection for fixed and random effects in linear mixed-effects models. *Biometrics* **66**(4), 1069–1077 (2010)
3. Catalano, P.J., Ryan, L.M.: Bivariate latent variable models for clustered discrete and continuous outcomes. *J. Am. Stat. Assoc.* **87**(419), 651 (1992)
4. Catalano, P.J., Scharfstein, D.O., Ryan, L.M., Kimmel, C.A., Kimmel, G.L.: Statistical model for fetal death, fetal weight, and malformation in developmental toxicity studies. *Teratology* **47**(4), 281–290 (1993)
5. Chen, J., Li, H.: Variable selection for sparse Dirichlet-multinomial regression with an application to microbiome data analysis. *Ann. Appl. Stat.* **7**(1), 418–442 (2013). <https://doi.org/10.1214/12-AOAS592>
6. Geys, H., Catalano, P., Faes, C.: Joint models for continuous and discrete longitudinal data. In: Verbeke, G., Davidian, M., Fitzmaurice, G., Molenberghs, G. (eds.) Longitudinal Data Analysis, volume 20085746 of Chapman & Hall/CRC Handbooks of Modern Statistical Methods, pp. 327–348. Chapman and Hall/CRC (2008)
7. Gueorguieva, R.: A multivariate generalized linear mixed model for joint modelling of clustered outcomes in the exponential family. *Stat. Model. Int. J.* **1**(3), 177–193 (2016)
8. Hartzel, J., Agresti, A., Caffo, B.: Multinomial logit random effects models. *Stat. Model. Int. J.* **1**(2), 81–102 (2016)
9. Iddi, S., Molenberghs, G.: A joint marginalized multilevel model for longitudinal outcomes. *J. Appl. Stat.* **39**(11), 2413–2430 (2012)
10. Kassahun, W., Neyens, T., Molenberghs, G., Faes, C., Verbeke, G.: A joint model for hierarchical continuous and zero-inflated overdispersed count data. *J. Stat. Comput. Simul.* **85**(3), 552–571 (2013)
11. Laird, N.M., Ware, J.H.: Random-effects models for longitudinal data. *Biometrics* **38**(4), 963 (1982)
12. Li, H.: Microbiome, metagenomics, and high-dimensional compositional data analysis. *Annu. Rev. Stat. Appl.* **2**(1), 73–94 (2015)
13. Liu, Q., Pierce, D.A.: A note on Gauss-Hermite quadrature. *Biometrika* **81**(3), 624 (1994)
14. Martin, I.: Mixed models for correlated compositional data: applied to microbiome studies in Indonesia. Ph.D. thesis, Leiden University (2019). <https://openaccess.leidenuniv.nl/handle/1887/79254>
15. Martin, I., Djuardi, Y., Sartono, E., Rosa, B.A., Supali, T., Mitreva, M., Houwing-Duistermaat, J.J., Yazdanbakhsh, M.: Dynamic changes in human-gut microbiome in relation to a placebo-controlled anthelmintic trial in Indonesia. *PLoS Negl. Trop. Dis.* **12**(8), e0006620 (2018)
16. Martin, I., Kaisar, M.M.M., Wiria, A.E., Hamid, F., Djuardi, Y., Sartono, E., Rosa, B.A., Mitreva, M., Supali, T., Houwing-Duistermaat, J.J., Yazdanbakhsh, M., Wammes, L.J.: The effect of gut microbiome composition on human immune responses: An exploration of interference by helminth infections. *Front. Genet.* **10**, 215 (2019a)
17. Martin, I., Uh, H.-W., Supali, T., Mitreva, M., Houwing-Duistermaat, J.J.: The mixed model for the analysis of a repeated-measurement multivariate count data. *Stat. Med.* **38**(12), 2248–2268 (2019b)
18. McCulloch, C.: Joint modelling of mixed outcome types using latent variables. *Stat. Methods Med. Res.* **17**(1), 53–73 (2008)
19. Neuhaus, A., Augustin, T., Heumann, C., Daumer, D.: A review on joint models in biometrical research. *J. Stat. Theory Pract.* **3**(4), 855–868 (2009)
20. Rosenthal, M., Aiello, A.E., Chenoweth, C., Goldberg, D., Larson, E., Gloor, G., Foxman, B.: Impact of technical sources of variation on the hand microbiome dynamics of healthcare workers. *PloS One* **9**(2), e88999 (2014)

21. Schloss, P.D., Gevers, D., Westcott, S.L.: Reducing the effects of PCR amplification and sequencing artifacts on 16s rRNA-based studies. *PLoS One* **6**(12), e27310 (2011)
22. Stefanski, L.A.: Measurement error models. *J. Am. Stat. Assoc.* **95**(452), 1353–1358 (2000)
23. Tutz, G.: Regression for Categorical Data. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge and New York (2012)
24. Verbeke, G., Fieuws, S., Molenberghs, G., Davidian, M.: The analysis of multivariate longitudinal data: a review. *Stat. Methods Med. Res.* **23**(1), 42–59 (2014)
25. Wammes, L.J., Hamid, F., Wiria, A.E., May, L., Kaisar, M.M.M., Prasetyani-Gieseler, M.A., Djuardi, Y., Wibowo, H., Kruize, Y.C.M., Verweij, J.J., de Jong, S.E., Tsonaka, R., Houwing-Duistermaat, J.J., Sartono, E., Luty, A.J.F., Supali, T., Yazdanbakhsh, M.: Community deworming alleviates geohelminth-induced immune hyporesponsiveness. *Proc. Natl. Acad. Sci. U.S.A.* **113**(44), 12526–12531 (2016)
26. Wiria, A.E., Prasetyani, M.A., Hamid, F., Wammes, L.J., Lell, B., Ariawan, I., Uh, H.W., Wibowo, H., Djuardi, Y., Wahyuni, S., Sutanto, I., May, L., Luty, A.J.F., Verweij, J.J., Sartono, E., Yazdanbakhsh, M., Supali, T.: Does treatment of intestinal helminth infections influence malaria? background and methodology of a longitudinal study of clinical, parasitological and immunological parameters in Nangapanda, Flores, Indonesia (immunospin study). *BMC Infect. Dis.* **10**, 77 (2010)
27. Yang, Y., Kang, J.: Joint analysis of mixed Poisson and continuous longitudinal data with nonignorable missing values. *Comput. Stat. Data Anal.* **54**(1), 193–207 (2010)

Statistical Methods for Feature Identification in Microbiome Studies



Peng Liu, Emily Goren, Paul Morris, David Walker, and Chong Wang

1 Introduction

A microbiome refers to a community of microorganisms residing in a specific host or location [19]. In recent years, microbiome studies have been facilitated by advancements in next-generation sequencing (NGS) technologies and promoted by funding agencies such as the National Institutes of Health (NIH) [25]. As a result, numerous microbiome studies have been carried out.

Several methods have been applied to survey diverse microbes from a given sample. Whole-metagenome shotgun (WMS) sequencing and amplicon sequencing target DNA, whereas metatranscriptomics targets RNA [18]. In the case of amplicon sequencing, sequence reads can be clustered into operational taxonomic units (OTUs) at a fixed level of base pair similarity (e.g., 97%) [41] or enumerating unique denoised (e.g., error-corrected) sequences called exact amplicon sequence variants (ASVs) [5]. Both the OTU and ASV approaches produce a high-dimensional vector of nonnegative integer counts for each sample, which can be classified to known taxa [30]. For an experiment with multiple samples, these features form a matrix of counts representing abundances within a given sample. Table 1 shows an example consisting of m (features) by n (sample) matrix of ASV counts. WMS sequencing and metatranscriptomics also result in such data matrices that relate abundance of features (taxa or genes) to samples. In this chapter, the methods that we discuss do not depend on the approach used to obtain microbiome features, and hence, we refer to each variable (OTU, ASV, taxon, or gene) as a feature.

P. Liu (✉) · E. Goren · P. Morris · D. Walker · C. Wang
Iowa State University, Ames, IA, USA
e-mail: pliu@iastate.edu; emily.goren@gmail.com; psmorris@iastate.edu; dcwalker@iastate.edu; chwang@iastate.edu

Table 1 A microbiome data matrix. A total of m microbiome features are measured for each of the n samples. Each count measures the abundance of the corresponding feature in a given sample

Sample Feature \	1	2	3	...	n
1	5	3	10	...	7
2	5	23	4	...	2
3	43	41	36	...	25
:	:	:	:	:	:
m	4	4	2	...	1

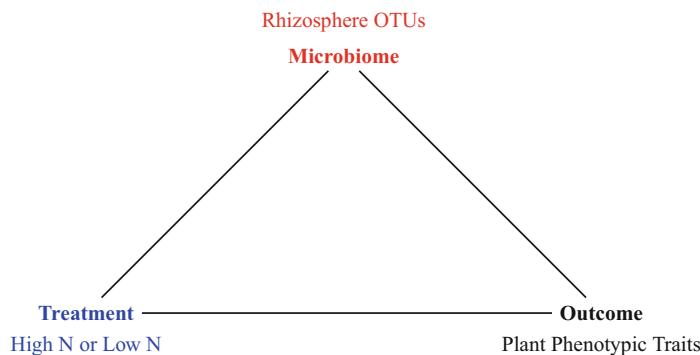


Fig. 1 An illustration of a hypothetical microbiome study, where complex interactions can exist between the treatment, the microbiome, and the outcome of interest

Microbiome studies try to understand how the microbiome functions and microbes influence the other parts of the biological system, such as metabolites. Take a microbiome study in agriculture as an example (Fig. 1). Plants grown under different environments (such as high- or low-nitrogen treatment) are sampled. Multiple variables are measured for each plant, including rhizosphere microbiome and other plant phenotypic traits or outcomes, to study the interaction between environment, microbiome, and plants. Scientists often aim to identify interesting microbial features for follow-up studies. Which features are considered “interesting” depends on the specific scientific questions being investigated. In this chapter, we focus on three target questions that define “interesting” features.

The first target question is which microbiome features are impacted by treatments or environmental conditions. For example, which rhizosphere microbes are affected by nitrogen levels in soil? Such analysis aims to identify features whose abundances change across treatments or conditions and has been called differential abundance analysis, analogous to differential expression analysis in gene expression studies.

The second target question is which microbiome features mediate treatment effects on an outcome. In the interplay of biological systems, it has been hypothesized that some microbes are affected by treatments and that the resulting changes in these microbes can influence the outcome. In Fig. 1, this corresponds to the path: Treatment → Microbiome → Outcome. For example, the abundances of nitrogen-fixing bacteria may be affected by nitrogen levels in the soil, and these bacteria help

utilize nitrogen and consequently affect the biomass of the plant. If this hypothesis is true, then identification of microbiome features that carry mediation effects will help develop targeted interventions that maximize the favorable treatment effect on the outcome. Statistical analysis that can identify mediation effects is called mediation analysis. Mediation analysis in microbiome studies is challenging because of the high dimensionality and sparsity of microbiome data.

The third target question is which microbiome features have an effect on an outcome, adjusting for confounders. In some microbiome studies, there are no particular treatments of interest, and the studies aim to identify microbiome features with an effect on an outcome. However, such studies often involve complex confounding arising from relationships between microbes, host, and environment [43]. There could exist confounding variables that affect both the outcome and at least some microbiome features. Selecting microbiome features with a relevant effect on the outcome requires statistical methodology that adjusts for the effects of such confounding variables. Unfortunately, the characteristics of microbiome data make it statistically challenging to adjust for confounding effects.

The next three sections present feature identification methods to answer each of the target questions described above. We also list the available R packages and provide example R code for the described methods. We conclude with a brief summary and discussion. For Bayesian feature/variable selection methods with microbiome data, readers are directed to chapter “Dirichlet-Multinomial Regression Models with Bayesian Variable Selection for Microbiome Data”.

2 Differential Abundance Analysis

Differential abundance analysis aims to identify which microbiome features are associated with variation in environmental, biological, or clinical conditions (Fig. 2). Hence, the null hypothesis of a differential abundance test is that treatments do

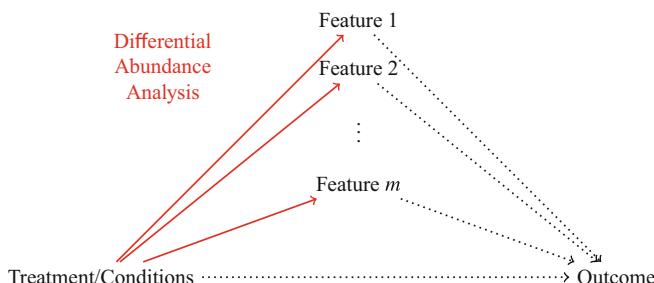


Fig. 2 Differential abundance analysis (solid lines) selects microbiome features whose abundance levels change across treatments or conditions. It only examines the relationship between treatments/conditions and microbiome features, but not the relationships involving other outcomes

not affect the mean abundance level. Approaches to differential abundance analysis treat microbiome feature counts either directly or by transformation, such as the compositional approach. The two conceptualizations are motivated by different aspects of the data. In the next two subsections, we consider several methods from both the compositional and count-based conceptualizations of the data. We explain the attributes of the data that motivate each concept and direct the reader to freely available R packages that implement the methods.

2.1 Compositional Methods

Due to the technical capacity of the sequencing technology, the total sum of feature counts in each sample is arbitrarily constrained, similar to classic compositional data [36].

Compositional approaches to differential abundance analysis use methods meant for compositional data, pioneered by Aitchison [1]. Aitchison's methods use one of the three log-ratio transformations to map compositions to Euclidean space. Let

$$X_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, m \quad (1)$$

represent the count for feature j within sample i . Dividing by the total sum for each sample transforms counts to proportions:

$$p_{ij} = \frac{X_{ij}}{\sum_{j=1}^m X_{ij}}. \quad (2)$$

The centered log-ratio (clr) transformation applies to the vector of sample proportions:

$$\text{clr}(\mathbf{p}_i) = \left[\log\left(\frac{p_{i1}}{g(\mathbf{p}_i)}\right), \dots, \log\left(\frac{p_{im}}{g(\mathbf{p}_i)}\right) \right]^T, \quad (3)$$

where $\mathbf{p}_i = (p_{i1}, \dots, p_{im})^T$ and $g(\mathbf{p}_i)$ denotes the geometric mean of the vector \mathbf{p}_i . Note that the covariance matrix of the transformed data will be singular. The additive log-ratio (alr) transformation is similar to the clr transformation, replacing $g(\mathbf{p}_i)$ with one of the components of \mathbf{p}_i in the denominator:

$$\text{alr}(\mathbf{p}_i) = \left[\log\left(\frac{p_{i1}}{p_{iD}}\right), \dots, \log\left(\frac{p_{i(m-1)}}{p_{iD}}\right) \right]^T. \quad (4)$$

The alr transform maps the vector of proportions to \mathbb{R}^{m-1} , with the choice of p_{iD} being arbitrary. The isometric log-ratio (ilr) transformation also maps compositional vectors to Euclidean space; it has the desirable property of isometry, but its

development is slightly more complicated. We refer the reader to [8] for a thorough development of the ilr transform.

Note that the log-ratio of 0 is undefined; since microbiome feature count tables contain many zeros, compositional approaches systematically remove or replace zero counts before transforming the data [36, 37].

Fernandes et al. [9] use a Monte Carlo sampling method to convert feature counts into proportions and simultaneously replace zeros with a small positive value. They then apply a clr transformation to the proportions. Aitchison shows standard multivariate hypothesis testing methods can be used on compositional data after applying the clr transformation [2]. Fernandes et al. [9] use both an unequal variances t -test and a Wilcoxon rank test for differential abundance of each feature between only two groups, with a Benjamini–Hochberg correction for multiple testing. Their method is implemented in the [ALDEx2](#) R package [10]. Although the paper only discusses two treatment comparisons, the R package includes functions for differential abundance analysis with more than two treatment groups.

Mandal et al. [22] replace zero counts with a small positive constant before converting counts into proportions and then apply an alr transformation to the proportions. Much like [9] above, Mandal et al. [22] use standard multivariate methods after transforming the data. They analyze the log-ratios using an ANOVA model and apply t - or F -tests for differences in abundance between treatment conditions. When the distributional assumptions are violated, Mandal et al. [22] substitute the Wilcoxon test or Kruskal–Wallis test in place of t - or F -tests. Note that for each feature j within each sample, the alr transformation generates $m - 1$ variables, using all other features $D \neq j$ in turn as the reference in the denominator (i.e., in the place of p_{iD} in Eq. (4)). Taking each pairwise comparison once over all m features, there are a total of $\binom{m}{2}$ data sets generated by the alr transformation. Each is used to test equality of means across treatment conditions. Mandal et al. [22] use either the Benjamini–Hochberg procedure to adjust all $\binom{m}{2}$ p -values or a multiple-comparison adjustment of their own devising. They then declare feature j is differentially abundant if W_j , the number of rejected null hypotheses involving feature j , is bigger than a threshold. The threshold can be $m - 1$ or $m - 2$, or a threshold determined by the empirical distribution of W_j .

Kaul et al. [17] extend the work of [22] to model classes of zeros in the data that originate by different mechanisms. After log-ratio transformation, they model the transformed data as a mixture of normal distributions. The method is implemented in the R package [ANCOM II](#). This implementation includes functions for handling additional covariates and mixed-effects models.

Methods that transform feature counts into proportions treat the data as relative rather than absolute. These methods might not be categorized as compositional if they do not follow the analytical steps developed by Aitchison. In Table 2, we label this approach “proportional.” Peng et al. [27] represent the proportions with a zero-inflated beta regression model. To test differential abundance between groups, they apply a likelihood-ratio test to the regression coefficients. Their method is implemented in the package [ZIBseq](#). Chen and Li [7] also use a zero-inflated beta regression model (implemented in [ZIBR](#)), with additional components to handle

Table 2 The list of differential abundance analysis methods described in Sect. 2

Feature type	R Package ^a	Model	Citation
Compositional	ALDEx2	ANOVA	[9]
	ANCOM II	ANOVA	[17]
	miLineage	ZIGDM ^b	[34]
Proportional	ZIBseq	Zero-inflated beta	[27]
	ZIBR	Zero-inflated beta	[7]
Count	DESeq	Negative binomial	[3]
	edgeR	Negative binomial	[29]
	metagenomeSeq	Zero-inflated Gaussian	[26]
	BhGLM	Zero-inflated negative binomial	[47]

^a Hyperlinks to the R packages are included

^b ZIGDM stands for zero-inflated generalized Dirichlet-multinomial

repeated measures. Tang and Chen [34] do not transform the data to proportions and instead model the counts using a zero-inflated generalized Dirichlet-multinomial (ZIGDM) model. The structure of this model represents the compositionality of the data, and score tests on transformations of the model parameters can test differential abundance. Functions to apply the method are available in the [miLineage](#) package.

2.2 Count-Based Methods

Although microbiome data has some of the attributes of compositional data, it is not perfectly compositional. Classic compositional data vectors represent portions of a whole. The total sum of the components is not meaningful, and only the relative difference between components matters [36]. For truly compositional data, the vectors (2, 1) and (2000, 1000) represent the same information: *only* that the first and second components are present in the ratio 2 : 1. For microbiome data, the size of the counts also contains information about the *reliability* of the ratio. Larger counts are more likely to closely match the true ratio in the sample [44].

Count-based methods address important attributes of microbiome feature counts: overdispersion and zero inflation. Using a negative binomial model can account for overdispersion. Zero-inflated and zero-hurdle models have been used to handle the zero inflation.

Gene expression data produced by RNA-sequencing (RNA-seq) technologies and microbiome count data have many similarities. Some of the first popular statistical methods and software packages used for differential abundance analysis were originally developed to identify differentially expressed genes in RNA-seq data. These methods can be applied directly to microbiome data, by treating microbiome feature counts as gene expression counts. McMurdie and Holmes [24] advocated the adoption of these methods in microbiome studies. [DESeq](#) [3] and

`edgeR` [29] are two examples of methods developed for RNA-seq data that have since been applied to microbiome data. Both use a negative binomial model for feature counts. They include several hypothesis testing procedures for both two-group and multiple-group comparisons.

Methods adapted from gene expression studies do not model the zero inflation that is typical of microbiome feature counts. Paulson et al. [26] use a zero-inflated normal model to represent the log of microbiome feature counts. This model is a mixture of a point mass at zero and a normal distribution. The method is implemented in `metagenomeSeq`. Zhang et al. [47] explore a zero-inflated negative binomial model for differential abundance testing, in the context of a generalized linear model. They develop an expectation–maximization (EM) algorithm to find the maximum likelihood estimator (MLE) of their model parameters and use these estimates and the asymptotic normality of the MLE to test for differential abundance. The method is implemented in the R package `BhGLM`. Jonsson et al. [16] use a zero-inflated overdispersed Poisson distribution to model feature counts in a Bayesian framework. They demonstrate that correctly modeling the zero inflation of microbiome data increases the power of differential abundance methods.

2.3 Additional Notes

In Table 2, we list the differential abundance analysis methods discussed in this section, including hyperlinks to corresponding R packages. Xia and Sun [43] review some methods for differential abundance analysis. Weiss et al. [40] also compare several differential abundance analysis methods using both real and simulated datasets; they find that no single method outperforms the others across a variety of settings.

Count-based and compositional approaches usually include preparatory steps before fitting the model or testing for differential abundance. Filtering out features with low counts across a large fraction of samples is commonplace. Count-based approaches usually normalize across samples; many normalization methods have been developed. Weiss et al. [40] also compare the performance of normalization methods, again returning nuanced answers: more than one approach has merit. They find evidence that rarefaction is useful and appropriate under some conditions; see also [24] and [23].

There are several packages that are not purpose-built for differential abundance analysis but may be helpful. The R package `compositions` has tools for analysis of compositional data; `phyloseq` has many tools for working with microbiome data; `pscl` implements several zero-inflated and zero-hurdle models—[44] offer a tutorial in fitting these models to microbiome data using the package. `DATest` provides tools to side-by-side test the performance of many differential abundance methods with user-provided data.

3 Mediation Analysis

While the literature on differential abundance analysis is well developed, many treatment–microbiome–outcome relationships fall outside of its scope. One such relationship is when treatment (exposure) affects the outcome indirectly through microbiome features. Mediation analysis allows for the examination of these indirect effects, which are distinct from the direct effect of treatment on the outcome that is not transmitted through the microbiome. In order for a feature to have a mediation effect, treatment must affect the feature and the change in the feature must result in an effect on the outcome. Figure 3 visualizes such pathways of feature-wise indirect effects: Treatment \rightarrow Feature \rightarrow Outcome. As an example, the diabetes treatment metformin contributes to changes in the gut microbiome, and these changes enhance the effects of the treatment above and beyond its direct effect [20, 42].

Mediation analyses have been implemented across a wide range of non-microbiome fields since [4] proposed a procedure to test for a single mediator in the context of psychology. Unfortunately, the methods that are appropriate for use with microbiome data are limited because of challenges inherent to the structure of the data. As described in previous sections, microbiome data are often high-dimensional, sparse, and formatted as either integer counts or compositions. Most existing methods deal with a small number of mediating features [15, 38]. While [46] and [14] propose methods for analyzing high-dimensional mediators, they both assume that the mediators are continuous, making the methods unsuitable for handling sparse count or compositional microbiome data.

The literature on mediation analysis for microbiome features is still in its infancy. The number of methods that have been developed is quite small, and there is not yet consensus on which methods are likely to perform well. Thus, rather than review any methods in detail, we provide a brief overview of each method and note when its use is appropriate. We structure the section around several broad categories of methods. Global methods test for the presence of a mediation effect but do not

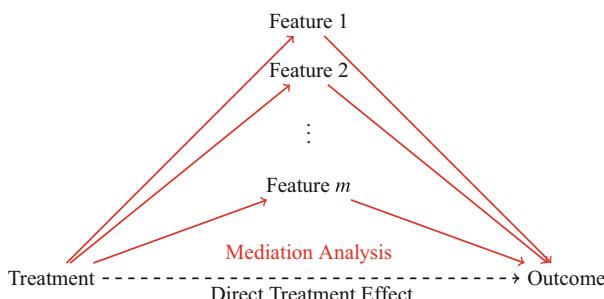


Fig. 3 Mediation analysis examines the indirect effects of treatment on the outcome through the microbiome. To determine whether a feature has a mediation effect, a method must consider both the effect of the treatment on the feature and the effect of the feature on the outcome

allow for identification of specific features that contribute to that effect. Targeted methods allow inference to be made for a single feature of interest while controlling for the other features. However, they do not provide any information on nontargeted features. Lastly, feature identification methods, the main focus of this section, can identify individual mediators through inference at the feature level. It is worth noting that these categories are not mutually exclusive. For example, several of the feature identification methods also test for the presence of a mediation effect at the global level. Table 3 groups methods by category and assumed feature type, as well as highlights when R packages or code are available to implement each method.

Global methods have been developed for use with count and compositional features. Zhang et al. [48] propose a nonparametric omnibus test for the presence of a mediation effect that utilizes phylogenetic and non-phylogenetic distance metrics simultaneously. The use of multiple distance metrics allows it to detect mediation effects of different forms. Hamidi et al. [12] also develop a method that tests for the presence of a mediation effect using a distance-based approach. The method relies on the calculation of pairwise distance matrices for the potentially multivariate exposures, mediators, and outcomes, where the distance measures are chosen according to the structure of the data and the problem at hand. Unlike the method proposed by [48], the method does not pool across multiple distance measures, making the choice of distance metric a key part of the analysis. Both methods are flexible in terms of the feature type.

Zhang et al. [49] present a targeted method for use with compositional features. They acknowledge the difficulties inherent to analyzing compositional data, specifically that any selection of $m - 1$ features may contain information on all m features because of the unit-sum constraint. To overcome this issue, they transform the m -dimensional microbiome data using the isometric log-ratio technique, which brings the data into Euclidean space of dimension $m - 1$. The targeted feature must be placed in the first column of the composition when performing the transformation, as the transformation is position-dependent. Parameters are estimated using ordinary least squares and the de-biased LASSO, and they develop what they denote a joint significance test for the mediation effect of the targeted feature that incorporates both the treatment–feature and feature–outcome relationships while controlling for the other features.

Table 3 The list of mediation analysis methods described in Sect. 3

Approach	Feature type	R Package ^a	Citation
Global	Count ^b	MedTest	[48]
	Count ^b	MODIMA	[12]
Targeted	Compositional	THIMA	[49]
Feature selection	Compositional	ccmm	[32]
	Compositional	SparseMCM	[39]
	Count		[6]

^a Hyperlinks to the R packages are included

^b Other feature types can be used

The first feature identification method we detail was proposed by [32]. They develop a compositional mediation model (CMM) for estimating the direct effect of treatment on the outcome and the component-wise indirect effects of high-dimensional compositional mediators, assuming a continuous outcome. The approach relies on compositional algebra to jointly estimate the effect of treatment on the features and linear log-contrast regression to estimate the effects of treatment and the microbiome features on the outcome. CMM tests global null hypotheses of no total mediation effect and no component-wise mediation effect for any feature using either a parametric or bootstrap approach. Confidence intervals for individual features can be calculated when using the bootstrap approach. The R package `cmmm` implements CMM.

Wang et al. [39] propose their Sparse Microbial Causal Mediation Model (SparseMCMM) to estimate the direct effect of treatment on the outcome and indirect effects at both the overall and feature levels for high-dimensional compositional mediators. Assuming a binary treatment and a continuous outcome, they use Dirichlet regression to estimate the effect of the treatment on the microbiome and linear log-contrast regression to estimate the effects of the treatment, the microbiome, and the treatment–microbiome interactions on the outcome. To account for the high dimensionality of microbiome data, they utilize regularization techniques to simultaneously estimate parameters and identify the features that serve as mediators. The method also tests global null hypotheses of no overall mediation effect and no component-wise mediation effects, using permutation to estimate significance. The authors provide the R package `SparseMCMM` as a means to implement their method.

Carter et al. [6] present a method named Nonparametric Entropy Mediation (NPEM) that can recognize nonlinear or non-additive relationships and does not assume specific data types for the multivariate exposures or the response variable. NPEM utilizes concepts from information theory to quantify the relevant associations and performs estimation through kernel density. They propose two approaches for testing to overcome issues that arise with kernel density estimation with sparse data. The first approach treats the features as counts, while the second decomposes the microbiome data into presence–absence and non-zero counts. Each approach tests for a mediation effect at the individual feature level, where a mediation effect means significant relationships between at least one of the exposures and the feature as well as the feature and the outcome.

In conclusion, the identification of individual microbiome features that mediate a treatment effect is particularly valuable, as it can allow for a more robust understanding of treatment–microbiome–outcome relationships. However, the development of feature identification methods is especially challenging. As more research is conducted, we hope to see additional methods proposed and a consensus on performance start to form.

4 Feature Identification Adjusting for Confounding

In this section, we consider identifying microbiome features with an effect on an outcome when there is no interest in the role of a treatment or other non-microbiome factors. As a motivating example, we consider identifying rhizosphere microbiome features that have an effect on plant biomass (Fig. 4). However, the study design may include a confounding variable, nitrogen fertilizer, that affects both the outcome and at least some microbiome features. In the context of studies aimed at identifying the role of the microbiome, a confounding variable or confounder is a variable correlated (either positively or negatively) with both the microbiome and an outcome of interest. Due to this relationship, studying the effect of the microbiome on the outcome while ignoring the confounder may not reflect the actual role of the microbiome. Experimental design techniques that control for confounding include randomization, restriction, and matching. We refer the reader to [13] for complete coverage of these methods. However, it is impractical or even impossible to use these methods in microbiome studies. Instead, statistical methods to adjust for potentially confounding variables may be used when the goal of analysis is to identify relevant microbiome features. The two most common options for confounding adjustment are through covariate adjustment and standardization.

4.1 Covariate Adjustment

Regression analysis, potentially with penalization for variable selection, has been used to analyze an outcome of interest modeled as a function of microbiome features [21, 28, 31, 45]. Certain confounding relationships can be appropriately handled

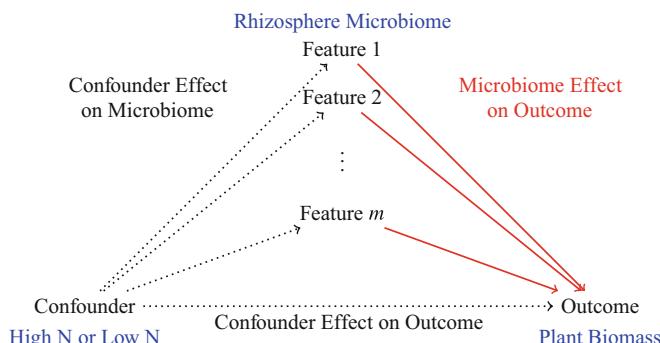


Fig. 4 When a confounder impacts both the microbiome and an outcome, accurately selecting relevant microbiome features requires accounting for the confounding effect. Section 4 mentions an agricultural microbiome project that aims to identify rhizosphere microbiome features that have an effect on plant biomass with nitrogen (N) level (high or low) as a confounder in this study

through inclusion as an additional covariate in a regression model. Let l_i be the value of the confounder. For example, suppose $l_i = 0$ when nitrogen level is high and $l_i = 1$ when nitrogen level is low. Let Y_i denote the outcome (such as biomass) for observation i , $i = 1, 2, \dots, n$, x_{ij} be the observed (possibly transformed) value for microbiome feature j in sample i , and $\mathbf{x}_i = (x_{i1}, \dots, x_{im})^T$ for the m -dimensional microbiome data vector of sample i . Including the confounder as an additional predictor in a generalized linear regression model is represented by

$$g\{\mathbb{E}(Y_i|\mathbf{x}_i, l_i)\} = \beta_0 + \sum_{j=1}^m \beta_j x_{ij} + \beta_\ell l_i, \quad (5)$$

where $g\{\mathbb{E}(Y_i)\}$ is the link function for the generalized linear model and β_ℓ is the parameter capturing the effect of the confounder. When a count-based approach for microbiome data is used, x_{ij} represents the normalized level of feature j for observation i . Compositional approaches may replace the normalized counts with a clr or log transformation and impose constraints on the microbiome regression coefficients. A detailed introduction to variable selection methods for microbiome compositional regression is provided by [33]. When using a penalized estimation approach such as LASSO [35] for the microbiome feature effects $\beta_1, \beta_2, \dots, \beta_m$, the confounder effect β_ℓ can be left unpenalized to ensure full adjustment. The set of selected microbiome features that are relevant with the outcome are those with $\beta_j \neq 0$.

A covariate adjustment approach through regression of the form in Eq. (5) offers flexibility in the values of the confounder: it can be discrete or continuous. Additional terms can be used for multiple confounders or multiple levels of a discrete confounder through dummy variables. However, such an approach cannot account for interaction effects between the microbiome and the confounder(s). With microbiome features often of high dimensionality ($m \gg n$), including an interaction term between the confounder and each feature is generally not feasible. Further, when a confounder affects many microbiome features, there will be confounder-induced marginal correlation between them, as illustrated in the next example (Fig. 5), which hinders the performance of variable selection methods. To overcome these challenges, the next section covers model-based standardization for a categorical confounder.

No single R package performs the analysis we discussed in this subsection directly. Hence, we provide R code to illustrate how to perform the covariate adjustment with the LASSO. Below is an example with $n = 200$ observations, half of which have the presence of a binary confounder $l_i = 1$, and $m = 50$ OTUs. We use the packages **HDeconometrics** to implement the LASSO penalty and **GGally** for correlation visualization.

First, a LASSO penalty is applied to the normalized and scaled OTUs, but not the confounder. Using BIC to select the penalty parameter, no microbiome features are selected. Figure 5 generated by the code below shows Spearman's rank correlation

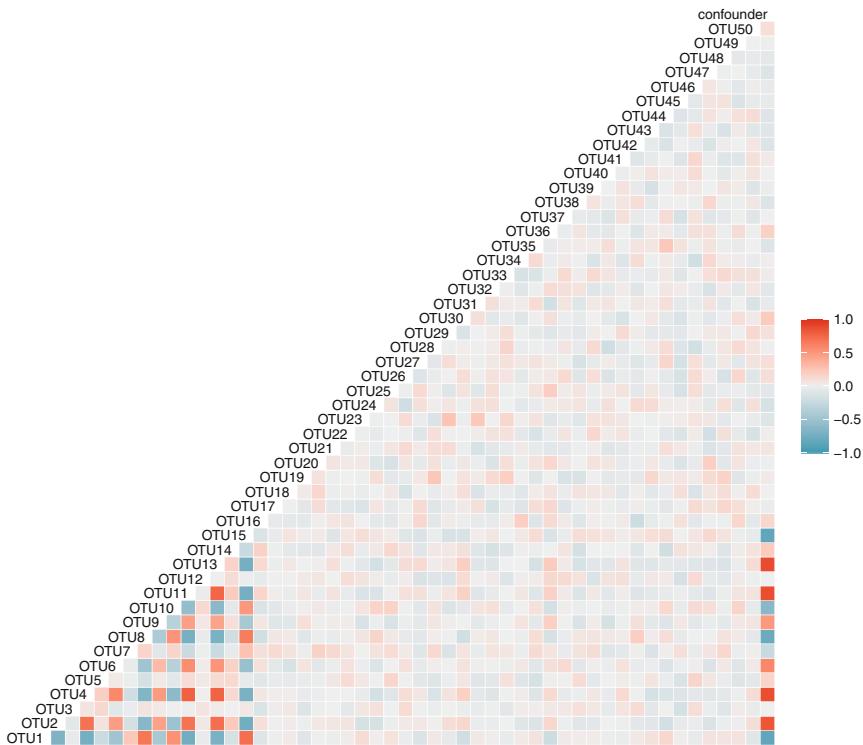


Fig. 5 Spearman's rank correlation between the (pairwise) OTUs and the confounder for the data in the R code example in Sect. 4.1

between the (pairwise) OTUs and the confounder. It shows that almost a third of the 50 OTUs are highly correlated with the confounder.

```
> # Number of observations
> n <- nrow(X)
> n
[1] 200
>
> # Number of observations per level of confounder
> table(confounder)
confounder
  0    1
100 100
>
> # Number of microbiome features
> m <- ncol(X)
> m
[1] 50
>
```

```

> # Summary of outcome
> summary(Y)
  Min. 1st Qu. Median     Mean 3rd Qu.      Max.
-0.5368  0.3753  2.7631  2.7127  5.0967  5.8172
>
> # Fit LASSO model with penalty not applied to confounder
> design_matrix <- cbind(scale(X), confounder)
> library(HDeconometrics)
> fit <- ic.glmnet(
+   x = design_matrix, y = Y,
+   crit = "bic", #choose penalty parameter using BIC
+   standardize = FALSE, #already scaled OTUs
+   penalty.factor = c(rep(1, m), 0)
+   # no variable selection for confounder
+ )
> names(which(fit$coefficients != 0))
[1] "(Intercept)" "confounder"
> # The above results indicate no features were selected,
  just the confounder
>
> # the correlation between many features and confounder is high
> library(GGally)
> ggcov(design_matrix, method = c("everything", "spearman"))

```

4.2 Model-Based Standardization

Model-based standardization estimates a population-averaged effect by performing estimation stratified by a categorical confounder and then standardizing the resulting estimate to the distribution in the population of interest. It requires the confounder l_i to be categorical with a finite number of levels, each represented sufficiently in the study of n observations. Denote the set of confounder values by \mathcal{L} . The observations are split into strata within which the confounder is equal. As a result, the value of the confounder does not vary within the group, and conditional on the confounder, the relationship between the microbiome and the outcome is unconfounded.

For each stratum $l \in \mathcal{L}$, the observations with that level of the confounder are used to estimate stratum-specific effects $\beta^l = (\beta_1^l, \beta_2^l, \dots, \beta_m^l)^T$ in the generalized linear model, as shown below.

$$g\{\mathbb{E}(Y_i | \mathbf{x}_i, l_i = l)\} = \beta_0 + \sum_{j=1}^m \beta_j^l x_{ij}. \quad (6)$$

This produces a set of estimates, $\hat{\beta}^l$, for each level of the confounder that can be standardized to a population of interest according to the distribution of the confounder. This standardized estimate is population averaged and computed by

$$\hat{\boldsymbol{\beta}} = \sum_{l \in \mathcal{L}} \hat{\boldsymbol{\beta}}^l \Pr(L = l), \quad (7)$$

where $\Pr(L = l)$ is the probability that the confounder takes the value l in the population of interest. Further details of this approach are presented in [11].

Below we revisit the example shown in Sect. 4.1 but perform a standardization approach by fitting a regression model with a LASSO penalty separately to each level of the confounder. A total of nine OTUs are selected.

```
> # Fit LASSO model to each confounder stratum
> coefs <- sapply(unique(confounder), function(l) {
+   ic.glmnet(
+     x = X[confounder == l, ], y = Y[confounder == l],
+     crit = "bic")$coefficients
+ })
> # Use row means to standardize since the confounder
> # is equally split over the groups
> standardized_coefficients <- rowMeans(coefs)
> names(which(standardized_coefficients != 0))
  "(Intercept)" "OTU1"          "OTU2"          "OTU3"
  "OTU6"          "OTU16"         "OTU17"         "OTU18"
  "OTU20"         "OTU28"
>
```

5 Summary

Feature identification or variable selection is a common problem in the analysis of high-dimensional omics data, including microbiome data. All omics data exhibit the so-called large m (dimension of variables), small n (sample size) problem that poses challenges. Some methods evaluate one feature at a time and then control multiple testing errors such as the differential abundance analysis discussed in Sect. 2. However, when treating omics data as covariates as involved in Sects. 3 and 4, variable screening and/or variable selection methods are in need. For microbiome data, an additional challenge is the sparsity and compositionality. We discussed issues related to compositional data analysis to some extent, and other chapters in this book also mentioned this challenge.

In this chapter, we cover statistical methods for microbiome feature identification that address three target questions. These questions arise in relatively simple settings common in microbiome studies that involve treatments (or environmental conditions or exposures), microbiome, and some outcomes of interest. Methods for these types of analyses are actively being developed as seen in the recent literature.

With advancements in technology, it is easier than ever to collect large amounts of data of different types, such as phenome, metabolome, transcriptome, and micro-

biome data, from a single study. In such studies, different parts of the biological systems are being measured. Hence, the three-way relationships in Fig. 1 can be expanded to multi-way relationships taking other omics data into consideration. Integrating multiple omics data can provide a holistic view of the biological systems under study. However, the computational methods are not yet able to efficiently utilize all available data sources to dissect complex biological processes and predict outcomes of interest with high precision. Integrating different omics data and identifying features that play important roles in the biological systems are challenging. Going forward, we expect the development of more system-level analysis that will likely involve machine learning and feature selection methods.

References

1. Aitchison, J.: *The Statistical Analysis of Compositional Data*. Chapman & Hall/CRC, Boca Raton (1986)
2. Aitchison, J.: *Principles of Compositional Data Analysis*. Lecture Notes-Monograph Series, pp. 73–81. Euclid, Durham (1994)
3. Anders, S., Huber, W.: Differential expression analysis for sequence count data. *Genome Biol.* **11**, R106 (2010)
4. Baron, R.M., Kenny, D.A.: The moderator-mediator variable distinction in social psychological research: conceptual, strategic, and statistical considerations. *J. Personal. Soc. Psychol.* **51**, 1173–1182 (1986)
5. Callahan, B.J., McMurdie, P.J., Holmes, S.P.: Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *ISME J.* **11**, 2639–2643 (2017)
6. Carter, K.M., Lu, M., Jiang, H., An, L.: An information-based approach for mediation analysis on high-dimensional metagenomic data. *Front. Genet.* **11**, 148 (2020)
7. Chen, E.Z., Li, H.: A two-part mixed-effects model for analyzing longitudinal microbiome compositional data. *Bioinformatics* **32**(17), 2611–2617 (2016)
8. Egozcue, J.J., Pawlowsky-Glahn, V., Mateu-Figueras, G., Barcelo-Vidal, C.: Isometric logratio transformations for compositional data analysis. *Math. Geol.* **35**(3), 279–300 (2003)
9. Fernandes, A.D., Reid, J.N., Macklaim, J.M., McMurrough, T.A., Edgell, D.R., Gloor, G.B.: Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16s rRNA gene sequencing and selective growth experiments by compositional data analysis. *Microbiome* **2**(1), 15 (2014)
10. Gloor, G.: ALDEX2: ANOVA-like differential expression tool for compositional data. *ALDEX Manual Mod.* **20**, 1–11 (2015)
11. Goren, E., Wang, C., He, Z., Sheflin, A.M., Chiniquy, D., Prenni, J.E., Tringe, S., Schachtman, D.P., Liu, P.: Feature selection and causal analysis for microbiome studies in the presence of confounding using standardization. *BMC Bioinformatics*, accepted (2021)
12. Hamidi, B., Wallace, K., Alekseyenko, A.V.: MODIMA, a method for multivariate omnibus distance mediation analysis, allows for integration of multivariate exposure-mediator-response relationships. *Genes* **10**, 524 (2019)
13. Hernán, M.A., Robins, J.M.: *Causal Inference: What If*. Boca Raton: Chapman & Hall/CRC (2020)
14. Huang, Y.-T., Pan, W.-C.: Hypothesis test of mediation effect in causal mediation model with high-dimensional continuous mediators. *Biometrics* **72**, 402–413 (2016)
15. Imai, K., Keele, L., Tingley, D.: A general approach to causal mediation analysis. *Psychol. Methods* **15**, 309–334 (2010)

16. Jonsson, V., Österlund, T., Nerman, O., Kristiansson, E.: Modelling of zero-inflation improves inference of metagenomic gene count data. *Stat. Methods Med. Res.* **28**(12), 3712–3728 (2019)
17. Kaul, A., Mandal, S., Davidov, O., Peddada, S.D.: Analysis of microbiome data in the presence of excess zeros. *Front. Microbiol.* **8**, 2114 (2017)
18. Knight, R., Vrbanac, A., Taylor, B.C., Aksenen, A., Callewaert, C., Debelius, J., Gonzalez, A., Koscielak, T., McCall, L.-I., McDonald, D., et al.: Best practices for analysing microbiomes. *Nat. Rev. Microbiol.* **16**, 410–422 (2018)
19. Lederberg, J., Mccray, A.T.: ‘Ome sweet’ omics—a genealogical treasury of words. *The Scientist* **15**(7), 8 (2001)
20. Li, H.: Statistical and computational methods in microbiome and metagenomics. In: Balding, D.J., Moltke, I., Marioni, J., Cannings, C., Bishop, M. (eds.) *Handbook of Statistical Genomics*, vol. 1, 4th edn., chap. 35, pp. 977–996. Wiley, Hoboken, NJ (2019)
21. Lu, J., Shi, P., Li, H.: Generalized linear models with linear constraints for microbiome compositional data. *Biometrics* **75**(1), 235–244 (2019)
22. Mandal, S., Van Treuren, W., White, R.A., Eggesbø, M., Knight, R., Peddada, S.D.: Analysis of composition of microbiomes: a novel method for studying microbial composition. *Microb. Ecol. Health Dis.* **26**(1), 27663 (2015)
23. McKnight, D.T., Huerlimann, R., Bower, D.S., Schwarzkopf, L., Alford, R.A., Zenger, K.R.: Methods for normalizing microbiome data: an ecological perspective. *Methods Ecol. Evol.* **10**(3), 389–400 (2019)
24. McMurdie, P.J., Holmes, S.: Waste not, want not: why rarefying microbiome data is inadmissible. *PLoS Comput. Biol.* **10**(4), e1003531 (2014)
25. NIH Human Microbiome Portfolio Analysis Team: A review of 10 years of human microbiome research activities at the US National Institutes of Health, fiscal years 2007–2016. *Microbiome* **7**, 31 (2019)
26. Paulson, J.N., Stine, O.C., Bravo, H.C., Pop, M.: Differential abundance analysis for microbial marker-gene surveys. *Nat. Methods* **10**(12), 1200 (2013)
27. Peng, X., Li, G., Liu, Z.: Zero-inflated beta regression for differential abundance analysis with metagenomics data. *J. Comput. Biol.* **23**(2), 102–110 (2016)
28. Randolph, T.W., Zhao, S., Copeland, W., Hullar, M., Shojai, A.: Kernel-penalized regression for analysis of microbiome data. *Ann. Appl. Stat.* **12**(1), 540–566 (2018)
29. Robinson, M.D., McCarthy, D.J., Smyth, G.K.: edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**(1), 139–140 (2010)
30. Schloss, P.D., Westcott, S.L.: Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Appl. Environ. Microbiol.* **77**(10), 3219–3226 (2011)
31. Shi, P., Zhang, A., Li, H.: Regression analysis for microbiome compositional data. *Ann. Appl. Stat.* **10**(2), 1019–1040 (2016)
32. Sohn, M.B., Li, H., et al.: Compositional mediation analysis for microbiome studies. *Ann. Appl. Stat.* **13**(1), 661–681 (2019)
33. Susin, A., Wang, Y., Lê Cao, K.-A., Calle, M. L.: Variable selection in microbiome compositional data analysis. *NAR Genom. Bioinf.* **2**(2), lqaa029 (2020)
34. Tang, Z.-Z., Chen, G.: Zero-inflated generalized Dirichlet multinomial regression model for microbiome compositional data analysis. *Biostatistics* **20**(4), 698–713 (2019)
35. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodological)* **58**(1), 267–288 (1996)
36. Van den Boogaart, K.G., Tolosana-Delgado, R.: *Analyzing Compositional Data with R*, vol. 122. Springer, New York (2013)
37. Van den Boogaart, K.G., Tolosana-Delgado, R., Bren, M.: Concepts for handling zeroes and missing values in compositional data. In: *Proceedings of IAMG*, vol. 6 (2006)
38. VanderWeele, T., Vansteelandt, S.: Mediation analysis with multiple mediators. *Epidemiol. Methods* **2**, 95–115 (2014)

39. Wang, C., Hu, J., Blaser, M.J., Li, H.: Estimating and testing the microbial causal mediation effect with high-dimensional and compositional microbiome data. *Bioinformatics* **36**, 347–355 (2019)
40. Weiss, S., Xu, Z.Z., Peddada, S., Amir, A., Bittinger, K., Gonzalez, A., Lozupone, C., Zaneveld, J.R., Vázquez-Baeza, Y., Birmingham, A., Hyde, E.R., Knight, R.: Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome* **5**(1), 27 (2017)
41. Westcott, S.L., Schloss, P.D.: De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. *PeerJ* **3**, e1487 (2015)
42. Wu, H., Esteve, E., Tremaroli, V., Khan, M.T., Caesar, R., Mannerås-Holm, L., Ståhlman, M., Olsson, L.M., Serino, M., Planas-Fèlix, M., Xifra, G., Mercader, J.M., Torrents, D., Burcelin, R., Ricart, W., Perkins, R., Fernàndez-Real, J.M., Bäckhed, F.: Metformin alters the gut microbiome of individuals with treatment-naïve type 2 diabetes, contributing to the therapeutic effects of the drug. *Nat. Med.* **23**, 850–858 (2017)
43. Xia, Y., Sun, J.: Hypothesis testing and statistical analysis of microbiome. *Genes Dis.* **4**(3), 138–148 (2017)
44. Xia, Y., Sun, J., Chen, D.-G.: Statistical Analysis of Microbiome Data with R. Springer, New York (2018)
45. Xiao, J., Chen, L., Johnson, S., Yu, Y., Zhang, X., Chen, J.: Predictive modeling of microbiome data using a phylogeny-regularized generalized linear mixed model. *Front. Microbiol.* **9**, 1391 (2018)
46. Zhang, H., Zheng, Y., Zhang, Z., Gao, T., Joyce, B., Yoon, G., Zhang, W., Schwartz, J., Just, A., Colicino, E., Vokonas, P., Zhao, L., Lv, J., Baccarelli, A., Hou, L., Liu, L.: Estimating and testing high-dimensional mediation effects in epigenetic studies. *Bioinformatics* **32**, 3150–3154 (2016a)
47. Zhang, X., Mallick, H., Yi, N.: Zero-inflated negative binomial regression for differential abundance testing in microbiome studies. *J. Bioinf. Genom.* **2**, 2 (2016b)
48. Zhang, J., Wei, Z., Chen, J.: A distance-based approach for testing the mediation effect of the human microbiome. *Bioinformatics* **34**(11), 1875–1883 (2018)
49. Zhang, H., Chen, J., Li, Z., Liu, L.: Testing for targeted mediation effect with application to human microbiome data. In: Statistics in Biosciences. Tianjin University, Tianjin (2019)

Statistical Methods for Analyzing Tree-Structured Microbiome Data



Tao Wang and Hongyu Zhao

1 Introduction

Advances in DNA sequencing technologies and data analysis tools have improved our ability to understand the taxonomic composition and biological function of complex and dynamic microbial ecosystems [42]. These experimental and computational methods have vastly increased the number of microbiome surveys being performed and generated massive amounts of data to be analyzed. For example, after careful design and sample collection, 16S ribosomal RNA (rRNA) gene sequencing uses primers that target highly variable regions of the 16S rRNA gene in order to provide a low-resolution view of microbial communities by quantifying relative abundances of microbial taxa and determining the phylogenetic placement of these taxa [22].

After quality control and data preprocessing, a typical microbiome dataset consists of a matrix that relates abundances of operational taxonomic units (OTUs, clustered sequences that represent bacteria types) to samples, a phylogenetic tree (constructed for a chosen gene, for example, the 16S rRNA gene) that reflects the evolutionary relationship of these OTUs, and metadata that provides information about the samples [8]. Considerable effort is then devoted to interrogating microbiome data to dissect relationships between hosts, microbes, and environmental

T. Wang

Department of Bioinformatics and Biostatistics, School of Life Sciences and Biotechnology,
Shanghai Jiao Tong University, Shanghai, China
e-mail: neowangtao@sjtu.edu.cn

H. Zhao (✉)

Department of Biostatistics, Yale School of Public Health, Yale University, New Haven,
CT, USA
e-mail: hongyu.zhao@yale.edu

factors [21]. Despite rapid progress in this field, analyzing microbiome data is complicated by several challenges [26, 65].

First, library sizes depend on the sequencing platform used and the number of samples that are multiplexed per run, and often vary over several ranges of magnitude across samples. Second, microbiome data are high dimensional and sparse, with OTU matrices containing a large number of columns (OTUs) and a high proportion of zeros. Third, the specimen is a fraction of the microbial ecosystem, and so the abundance data in a sample are compositional carrying only relative information. Finally, OTUs are phylogenetically related, and the phylogeny can and should be leveraged [35, 64].

Analyses of microbiome data can be performed either at overall community level or at individual taxon level. For example, one of the first analysis steps is the calculation of a measure of dissimilarity between samples, known as beta diversity. An ordination technique, such as principal coordinate analysis, is then applied to dissimilarity matrices to visualize differences and similarities between microbial populations. Distance metrics that capture the phylogenetic information or account for the compositional nature have been developed [30, 47]. Differential abundance testing, on the other hand, refers to a diverse set of methods for detecting microbial taxa that are significantly differentially abundant between groups of interest (e.g., healthy versus diseased or control versus treatment) [32, 39].

Taxon abundances in microbiome data analysis can be characterized as either input or output variables, and learning problems either supervised or unsupervised. For example, in the supervised learning domain, we may wish to fit a model that relates a phenotype of interest to microbial abundances in order to accurately predict the phenotype or better understand the relationship between the phenotype and the microbial taxa [20]. Alternatively, we may be interested in dissecting how environmental factors and host genetics jointly shape human gut microbiome [44]. Unsupervised learning, on the other hand, describes the situation in which there is no particular phenotype to predict, and one seeks to understand the relationships between the microbes or between the observations [16].

This chapter reviews statistical models for multivariate microbial counts, empirical Bayes estimation of microbial relative abundances, and regularization methods for subcomposition selection and dimension reduction in regression with compositional predictors, with an emphasis on how to address some of the aforementioned challenges, in particular the incorporation of the phylogeny into analyses.

2 Modeling Multivariate Count Data

Let $\mathbf{X} = (X_1, \dots, X_K)^\top$ denote the random vector of counts on K bacterial taxa or OTUs and $M = \sum_{k=1}^K X_k$ the total number of counts. One natural distribution for describing \mathbf{X} is the multinomial (MN) distribution, with size m (that is, by conditioning on $M = m$) and vector of probabilities $\mathbf{p} = (p_1, \dots, p_K)^\top$, $p_k > 0$, $\sum_{k=1}^K p_k = 1$. The probability mass function is

$$f_{\text{MN}}(\mathbf{x}; \mathbf{p}, m) = \frac{\Gamma(m+1)}{\prod_{k=1}^K \Gamma(x_k + 1)} \prod_{k=1}^K p_k^{x_k}, \quad (1)$$

where $\mathbf{x} = (x_1, \dots, x_K)^\top$ is a realization of X , $m = \sum_{k=1}^K x_k$, and $\Gamma(\cdot)$ is the gamma function.

2.1 Dirichlet-Multinomial Model

One problem with MN is its difficulty in modeling over-dispersion, which is a well-known characteristic of count data in microbiome studies. To account for over-dispersion, the standard convention is to assume that \mathbf{p} is random with some prior distribution. Let

$$\mathbb{S}^{K-1} = \left\{ \boldsymbol{\pi} = (\pi_1, \dots, \pi_K)^\top : \pi_k > 0, \sum_{k=1}^K \pi_k = 1 \right\}$$

denote the $(K - 1)$ -dimensional simplex. Then, \mathbb{S}^{K-1} is the support of \mathbf{p} . The most common and convenient prior for \mathbf{p} is the Dirichlet distribution. This distribution is indexed by a K -vector of positive scalars, $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_K)^\top$, $\alpha_k > 0$, and has a probability density function

$$f_D(\mathbf{p}; \boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K p_k^{\alpha_k - 1}. \quad (2)$$

The Dirichlet-multinomial (DM) distribution, also known as the Dirichlet compound MN distribution, results from calculating the joint distribution for X and \mathbf{p} and then integrating out \mathbf{p} :

$$\begin{aligned} f_{\text{DM}}(\mathbf{x}; \boldsymbol{\alpha}, m) &= \int_{\mathbb{S}^{K-1}} f_{\text{MN}}(\mathbf{x}; \mathbf{p}, m) \times f_D(\mathbf{p}; \boldsymbol{\alpha}) d\mathbf{p} \\ &= \frac{\Gamma(m+1)\Gamma(\alpha_+)}{\Gamma(m+\alpha_+)} \prod_{k=1}^K \frac{\Gamma(x_k + \alpha_k)}{\Gamma(x_k + 1)\Gamma(\alpha_k)}, \end{aligned} \quad (3)$$

where $\alpha_+ = \sum_{k=1}^K \alpha_k$.

To incorporate covariates into the model, [9] related the parameters α_k to a q -dimensional vector of covariates $\mathbf{z} = (z_1, z_2, \dots, z_q)^\top$ via a log-linear transformation

$$\log(\alpha_k) = \sum_{j=1}^q \beta_{kj} z_j,$$

for $k = 1, \dots, K$, where β_{kj} measures the effect of the j th covariate on the k th taxon. When q is small, maximum likelihood estimation and inference for this DM regression can be applied. When q is large, regularization is useful for reducing the variance of estimator and/or improving its interpretability. Chen and Li [19] developed a penalized likelihood method for parameter estimation and variable selection.

Let $\pi_k = \alpha_k/\alpha_+$, $\phi = 1/(1 + \alpha_+)$, and $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)^\top$. One can re-parameterize DM as

$$f_{\text{DM}}(\mathbf{x}; \boldsymbol{\pi}, \phi, m) = \frac{\Gamma(m+1)}{\prod_{k=1}^K \Gamma(x_k+1)} \frac{\prod_{k=1}^K \prod_{l=1}^{x_k} \{\pi_k(1-\phi) + (l-1)\phi\}}{\prod_{j=1}^m \{1-\phi + (j-1)\phi\}}. \quad (4)$$

Since $\boldsymbol{\pi} \in \mathbb{S}^{K-1}$, DM is MN augmented with one additional parameter ϕ . We call ϕ the over-dispersion parameter. When $\phi = 0$, DM reduces to MN. Using this re-parameterization, [24] developed multivariate methods for hypothesis testing and power calculations for comparing multiple groups of microbiome samples, and [50] proposed an adaptive likelihood-ratio test of independence between the microbial community composition and a many-valued or continuous phenotype.

It is known that the dependence structure permitted by MN and DM is limited: the components are constrained to be negatively correlated. This negative correlation is induced by the constant sum constraint (m for the multinomial distribution and 1 for the Dirichlet distribution). Due to similar habitats or symbiotic interactions, microbes may display positive associations. In the presence of both negative and positive correlations, MN and DM are not adequate for characterizing microbiome data. Furthermore, they ignore the fact that microbial taxa are related evolutionarily on a phylogenetic tree. To address these limitations, [60] proposed an extension of DM called the Dirichlet-tree multinomial (DTM) distribution. Rather than placing a single DM on all taxa, DTM consists of a collection of independent DMs, each corresponding to an internal node of the phylogenetic tree, as shown below.

2.2 Dirichlet-Tree Multinomial Model

Suppose that the evolutionary relationships among OTUs are encoded by a rooted tree $\mathcal{T} = (\mathcal{L}, \mathcal{I})$, where terminal nodes, or leaves, in \mathcal{L} correspond to OTUs, and internal nodes in \mathcal{I} represent bacterial taxa at different taxonomic levels. For simplicity, we assume that $\mathcal{L} = \{1, \dots, K\}$. Figure 1 shows two binary trees, each with $K = 4$ leaves.

For each internal node $v \in \mathcal{I}$, let C_v be the set of child nodes of v . For each v and $c \in C_v$, define $\delta_{vc}(l)$ to be 1, if there is a path from v to c to $l \in \mathcal{L}$, and 0, otherwise. Denote by $X_{vc} = \sum_{l \in \mathcal{L}} \delta_{vc}(l) X_l$ the count in the subtree indexed by $c \in C_v$. Similarly, $p_{vc} = \sum_{l \in \mathcal{L}} \delta_{vc}(l) p_l$. One attractive property of the MN distribution is that it can be factorized over \mathcal{T} . Specifically, let $b_{vc} = p_{vc} / \sum_{c \in C_v} p_{vc}$, $\mathbf{b}_v = (b_{vc}, c \in C_v)$, $\mathbf{X}_v = (X_{vc}, c \in C_v)$, and $M_v = \sum_{c \in C_v} X_{vc}$. Then,

$$f_{\text{MN}}(\mathbf{x}; \mathbf{p}, m) = \prod_{v \in \mathcal{I}} f_{\text{MN}}(\mathbf{x}_v; \mathbf{b}_v, m_v) = \prod_{v \in \mathcal{I}} \frac{\Gamma(m_v + 1)}{\prod_{c \in C_v} \Gamma(x_{vc} + 1)} \prod_{c \in C_v} b_{vc}^{x_{vc}}.$$

The appropriate prior for this re-parameterization is no longer a single global Dirichlet density, but rather a product of local Dirichlet densities, one at each internal node of the tree:

$$\prod_{v \in \mathcal{I}} f_{\text{D}}(\mathbf{b}_v; \boldsymbol{\alpha}_v) = \prod_{v \in \mathcal{I}} \frac{\Gamma(\sum_{c \in C_v} \alpha_{vc})}{\prod_{c \in C_v} \Gamma(\alpha_{vc})} \prod_{c \in C_v} b_{vc}^{\alpha_{vc}-1},$$

where $\boldsymbol{\alpha}_v = (\alpha_{vc} > 0, c \in C_v)$ is a vector of positive scalars.

The DTM distribution is then defined as the product of DM distributions that factorize over the tree

$$\begin{aligned} f_{\text{DTM}}(\mathbf{x}; \boldsymbol{\alpha}_v, v \in \mathcal{I}, m) &= \prod_{v \in \mathcal{I}} \int_{\mathbb{S}^{K_v-1}} f_{\text{MN}}(\mathbf{x}_v; \mathbf{b}_v, m_v) \times f_{\text{D}}(\mathbf{b}_v; \boldsymbol{\alpha}_v) d\mathbf{b}_v \\ &= \prod_{v \in \mathcal{I}} \frac{\Gamma(m_v + 1)\Gamma(\alpha_{v+})}{\Gamma(m_v + \alpha_{v+})} \prod_{c \in C_v} \frac{\Gamma(x_{vc} + \alpha_{vc})}{\Gamma(x_{vc} + 1)\Gamma(\alpha_{vc})}. \quad (5) \end{aligned}$$

Here, K_v is the number of children of v , and $\alpha_{v+} = \sum_{c \in C_v} \alpha_{vc}$. A special case of DTM, known as the generalized Dirichlet-multinomial (GDM) distribution, was developed by [10], when the tree structure is restricted to a binary cascade (right panel of Fig. 1).

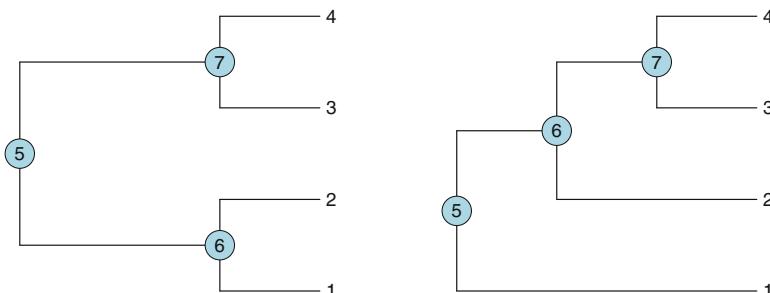


Fig. 1 Two binary trees, each with four leaves and three internal nodes

In the DTM regression model, the parameters $\{\alpha_v, v \in \mathcal{I}\}$ are related to the covariates $z = (z_1, z_2, \dots, z_q)^\top$ by the log link function

$$\log(\alpha_{vc}) = \sum_{j=1}^q \beta_{vcj} z_j,$$

for $v \in \mathcal{I}$ and $c \in C_v$, where β_{vcj} are regression coefficients. Wang and Zhao [60] introduced this model to study the effect of nutrient intake on gut microbiome and proposed a regularized likelihood approach for selecting relevant dietary nutrients and their associated taxa. Taking advantage of the correlated signals on the tree, [53] developed a phylogenetic scan test using the DTM model for investigating cross-group differences in microbiome compositions and applied it to identify bacterial taxa associated with diet habits.

2.3 Implementation and Illustration

We illustrate the use of the aforementioned methods by applying them to the COMBO dataset from a cross-sectional study on relating dietary habits to gut microbiome composition [66]. 98 healthy individuals were enrolled in this study, and their stool samples were collected. DNA samples were analyzed by the 454/Roche pyrosequencing of 16S rRNA gene segments of the V1–V2 region, and pyrosequences were processed by the QIIME pipeline [8]. More than 17,000 species-level OTUs (including the singletons) and a phylogenetic tree were produced. Clinical measurements, such as body mass index (BMI), were also collected.

We created a nontrivial set of OTUs by agglomerating closely related OTUs: all leaf nodes of the tree separated by a cophenetic distance (which is computed between pairs of tips from a phylogenetic tree using its branch lengths) smaller than 0.5 were agglomerated into one OTU [36]. For each merged group of OTUs, we chose the OTU with the highest abundance to represent it. After the processing steps, we were left with 98 individuals and 62 OTUs. We will be working with the `phyloseq` object. Note that program code for the analyses in this chapter can be found under <https://github.com/liudoubletian/phyloMDA>.

Load example data

```
# phyloseq: A tool to import, store, analyze,
# and display phylogenetic sequencing data
library(phyloseq); packageVersion("phyloseq")

# phyloMDA: An R package for phylogeny-aware
# microbiome data analysis
```

```
# install_github("liudoubletian/phyloMDA")
library(phyloMDA); packageVersion("phyloMDA")
(phyloseq.obj <- combo.phyloseq.obj)

# Plot the phylogenetic tree
plot_tree(phyloseq.obj, "treeonly",
          nodeplotblank, label.tips="taxa_names")
tree <- phy_tree(phyloseq.obj)

# Heatmap of microbial counts
plot_heatmap(phyloseq.obj,
             taxa.order=taxa_names(phyloseq.obj))
otu_tab <- t(phyloseq.obj@otu_table@Data)

# Metadata
metadata <- sample_data(phyloseq.obj)
```

The parameters of the DM distribution can be estimated using the method of moments or maximum likelihood [24]. Since the distributions placed on different internal nodes are independent, estimation of the parameters of the DTM distribution can be carried out separately and in parallel. Zhang et al. [68] proposed an iteratively reweighted Poisson regression method for maximum likelihood estimation for a class of regression models including the DM model. They also investigated testing and regularized estimation for these models.

Multinomial-logit regression

```
# MGLM: A package for multivariate response GLMs
library(MGLM); packageVersion("MGLM")

fit_mn <- MGLMfit(data=otu_tab, dist="MN")
fit_mn@logL # MN loglikelihood

sodium <- metadata$sodium
reg_mn <- MGLMreg(otu_tab~1+sodium, dist="MN")
reg_mn@logL # simple MN regression loglikelihood
```

Dirichlet-multinomial regression

```

fit_dm <- MGLMfit(data=otu_tab, dist="DM")
fit_dm@logL # DM loglikelihood

reg_dm <- MGLMreg(otu_tab~1+sodium, dist="DM")
reg_dm@logL # simple DM regression loglikelihood

# MGLMsparseereg and MGLMtune fit sparse regression
Nutrs <- metadata[, 18:37] # first 20 nutrients
Nutrs <- as.matrix(data.frame(Nutrs))
idx <- c(F, rep(T, dim(Nutrs)[2]))

sreg_dm <- MGLMsparseereg(otu_tab~1+Nutrs, dist="DM",
                           lambda=Inf, penalty="sweep", penidx=idx)
sreg_dm@logL

sreg_dm_tune <- MGLMtune(otu_tab~1+Nutrs, dist="DM",
                           penalty="sweep", penidx=idx)
sreg_dm_tune@select@logL

```

Dirichlet-tree multinomial regression

```

# library(phyloMDA); packageVersion("phyloMDA")

fit_dtm <- MGLMdtmpFit(otu_tab, tree)
Extract_logL(fit_dtm) # DTM loglikelihood

reg_dtm <- MGLMdtmpReg(otu_tab, sodium, tree)
Extract_logL(reg_dtm) # DTM regression loglikelihood

sreg_dtm <- MGLMdtmpSparseReg(otu_tab, Nutrs, tree,
                               lambda=Inf, penalty="sweep")
Extract_logL(sreg_dtm)

sreg_dtm_tune <- MGLMdtmpTune(otu_tab, Nutrs, tree,
                               penalty="sweep")
Extract_logL(sreg_dtm_tune)

```

3 Estimating Microbial Compositions

Although microbial community sequencing has expanded our knowledge of the role of microbes in human health and disease, the interpretation of microbiome data is complicated by two challenges [65]. First, library sizes depend on the sequencing platform used and the number of samples that are multiplexed per run, and often vary over several ranges of magnitude across samples. The large variability in sequence depth reflects deficiency of the sequencing process rather than true biological variation. Second, a specimen is just a fraction of the original ecosystem; hence, the total number of reads obtained for a sample is not itself informative, and the observed read counts in the sample provide only information about the relative abundances of OTUs in the specimen.

To deal with these challenges, microbiome data are often normalized by either a statistical or computational process prior to downstream analysis [37]. Two widely used normalization approaches are rarefying, which subsamples the data without replacement to uniform sequence depth, and total sum scaling (TSS), which divides read counts by the total count in each sample. Although rarefying is a primary method of choice in the researcher's toolkit, it throws away some data and thus is inadmissible. On the other hand, due to limited sequencing depth, undersampling, and DNA dropouts, most microbes are absent in the majority of samples, and hence, the OTU count matrix is sparse containing a high proportion of zeros [65]. Consequently, the relative abundances from TSS have excessive zeros, which can have an undesirable effect on downstream data analyses such as diversity estimation and log-like transformation (see Sect. 4).

3.1 Empirical Bayes Normalization

Consider a microbiome dataset with n samples and K OTUs. For the i th sample, let $\mathbf{x}_i = (x_{i1}, \dots, x_{iK})^\top$ be the vector of read counts of K OTUs and $m_i = \sum_{k=1}^K x_{ik}$ the total number of reads. There is a simple explanation for TSS. Suppose that X_i is a random draw from MN, with size m_i and probabilities p_{ik} , and \mathbf{x}_i is an instance of X_i . Then, the method of maximum likelihood yields the TSS normalization

$$\tilde{p}_{ik} = \frac{x_{ik}}{m_i}. \quad (6)$$

In other words, TSS is a model-based method. In the rest of this section, we show that DM is useful for data normalization from an empirical Bayes perspective [28].

As mentioned, one potential drawback of MN is that the estimated relative abundances $\tilde{p}_{ik} = 0$ for OTUs with zero counts. One way to overcome this problem, presented below, is to use a Bayesian approach by specifying a prior for the probability vector $\mathbf{p}_i = (p_{i1}, \dots, p_{iK})^\top$, calculating the posterior distribution of \mathbf{p}_i given \mathbf{x}_i , and then computing the posterior mean. Among all distributions on

the simplex \mathbb{S}^{K-1} , the Dirichlet distribution is popular mainly because, as a prior, it is conjugate to the MN.

Multiplying the MN distribution $f_{\text{MN}}(\mathbf{p}_i; m_i)$ with the Dirichlet prior $f_D(\boldsymbol{\alpha})$, we obtain the posterior distribution

$$f(\mathbf{p}_i; \mathbf{x}_i, \boldsymbol{\alpha}) \propto \prod_{k=1}^K p_k^{x_{ik} + \alpha_k - 1}.$$

We see that the posterior distribution again takes the form of a Dirichlet distribution, indexed by $\mathbf{x}_i + \boldsymbol{\alpha}$, confirming that the Dirichlet is indeed a conjugate prior for the MN. The posterior mean is given by

$$E(p_{ik} | \mathbf{x}_i, \boldsymbol{\alpha}) = \frac{x_{ik} + \alpha_k}{\sum_{l=1}^K (x_{il} + \alpha_l)}. \quad (7)$$

Since $\alpha_k > 0$, the estimated relative abundances are nonzero for all OTUs. Indeed, it is easy to check that the posterior mean is a weighted average of the TSS solution and the mean of the prior distribution:

$$E(p_{ik} | \mathbf{x}_i, \boldsymbol{\alpha}) = \frac{m_i}{m_i + \alpha_+} \tilde{p}_{ik} + \frac{\alpha_+}{m_i + \alpha_+} \pi_k,$$

where again $\alpha_+ = \sum_{k=1}^K \alpha_k$ and $\pi_k = \alpha_k / \alpha_+$. Put another way, we shrink the TSS estimates toward our knowledge about \mathbf{p}_i before we see the data.

In practice, the hyper-parameters α_k are unknown, and so the posterior mean cannot be used directly. Uniform priors, which assume that $\alpha_1 = \dots = \alpha_K$, are used in the literature [14, 34]. The mean vector of a uniform prior, $(1/K, \dots, 1/K)^\top$, is the center or neural element of \mathbb{S}^{K-1} with the Aitchison metric [40]. Nevertheless, we do not have to take this composition as the preferred shrinking point.

In the above derivation, the Bayesian approach is applied to single data points, but the observations may have much in common, and these similarities can be used to learn from the experience of others. In Sect. 2.1, we showed that the marginal distribution of X_i is DM, with the same set of parameters $\boldsymbol{\alpha}$ as the Dirichlet prior. We can estimate these parameters from OTU counts across samples by maximum likelihood, then plug-in the estimates into the prior distribution, and normalize the data using the posterior mean [28]. Let $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \dots, \hat{\alpha}_K)^\top$ be the maximum likelihood estimate under the DM model. Substituting it into (7) gives the empirical Bayes solution for normalization

$$\hat{p}_{ik} = \frac{x_{ik} + \hat{\alpha}_k}{\sum_{l=1}^K (x_{il} + \hat{\alpha}_l)}. \quad (8)$$

3.2 Phylogeny-Aware Normalization

We can further incorporate phylogeny into the normalization process [28]. First, the same technique applies to each internal node, giving a posterior distribution in the Dirichlet form

$$f(\mathbf{b}_v; \mathbf{x}_v, \boldsymbol{\alpha}_v) \propto \prod_{c \in C_v} b_{vc}^{x_{vc} + \alpha_{vc} - 1},$$

using the same notation as in Sect. 2.2. Here, for ease of notation, we omit the subscript i . Second, the posterior density function of \mathbf{p} given \mathbf{x} can be computed by a change of variables [11]. In particular, the posterior mean is

$$E(p_k | \mathbf{x}, \boldsymbol{\alpha}_v, v \in \mathcal{I}) = \prod_{v \in \mathcal{I}} \prod_{c \in C_v} \left\{ \frac{x_{vc} + \alpha_{vc}}{\sum_{c \in C_v} (x_{vc} + \alpha_{vc})} \right\}^{\delta_{vc}(l)}. \quad (9)$$

Finally, the Bayes estimator is itself being empirically estimated from the data by maximizing the marginal likelihood of the data, which are distributed as DTM. Let $\hat{\boldsymbol{\alpha}}_v = (\hat{\alpha}_{vc}, c \in C_v)$ be the maximum likelihood estimate. Substituting it into (9) leads to the phylogeny-aware normalization

$$\hat{p}_k = \prod_{v \in \mathcal{I}} \prod_{c \in C_v} \left\{ \frac{x_{vc} + \hat{\alpha}_{vc}}{\sum_{c \in C_v} (x_{vc} + \hat{\alpha}_{vc})} \right\}^{\delta_{vc}(l)}. \quad (10)$$

3.3 Statistical Analysis of Compositional Data

The relative abundances are compositional, since they sum up to one and are non-negative. More generally, a compositional data point, or composition for short, is a vector with strictly positive components whose sum is constant [1]. For simplicity, in this chapter, we restrict our discussion to compositions with a unit sum.

Microbiome data are compositional and should be treated as compositions. However, since compositions are constrained by the simplex, the analysis of compositional data using traditional tools can be misleading [17]. One way to address this issue is to use ratio transformations and then take the logarithm of these ratios, known as log-ratios. This is a reasonable strategy since compositional data quantitatively describe the parts of whole and contain only relative information between their components. The log-ratios are real numbers free of the unit-sum constraint, and they allow the application of standard methods that have been developed for real-valued data.

Often, the additive log-ratio (alr) and centered log-ratio (clr) transformations are used. Given a composition $\mathbf{p} = (p_1, \dots, p_K)^\top \in \mathbb{S}^{K-1}$, the alr transformation is

defined as the logarithm of the ratios of components over a given one, mapping \mathbf{p} to a vector in \mathbb{R}^{K-1} :

$$\text{alr}(\mathbf{p}) = \left(\log \frac{p_1}{p_K}, \dots, \log \frac{p_{K-1}}{p_K} \right)^\top.$$

Here, the last component is chosen by convention as the reference. The clr transformation is defined to be the log-ratios of \mathbf{p} over the geometric mean of \mathbf{p} :

$$\text{clr}(\mathbf{p}) = \left(\log \frac{p_1}{\sqrt[K]{\prod_{k=1}^K p_k}}, \dots, \log \frac{p_K}{\sqrt[K]{\prod_{k=1}^K p_k}} \right)^\top.$$

Important properties of these transformations include scale invariance and subcompositional consistency [12].

3.4 Implementation and Illustration

We illustrate the usefulness of the empirical Bayes method by applying it to the COMBO dataset. We are interested in identifying the taxa that are associated with BMI. We categorized BMI as normal weight, overweight, and obese and focused on the normal weight and obese individuals (in the next section, we discuss regression problems with compositional predictors and a continuous response).

Empirical Bayes normalization

```
# library(phyloMDA); packageVersion("phyloMDA")

eBay.comps <- eBay_comps(otu_tab, prior="Dirichlet")

eBay.tree.comps <- eBay_comps(otu_tab,
                                prior="Dirichlet-tree", tree=tree)
```

Log-ratio transformations

```
# library(phyloMDA); packageVersion("phyloMDA")

eBay.comps.alr <- alr_trans(eBay.comps)
```

```
eBay.comps.clr <- clr_trans(eBay.comps)
```

We used the empirical Bayes method to estimate relative abundances from the OTU count matrix. We then performed differential abundance testing between normal weight and obese subjects. For comparison, we present the corresponding results of applying other normalization and detection methods. From Fig. 2, we see that the empirical Bayes method detected more differentially abundant OTUs than others. At the phylum level, these OTUs belonged to *Bacteroidetes* and *Firmicutes* (see Table 1). It has been experimentally shown that in humans on weight-reduction diets, the decrease in *Bacteroidetes* was accompanied by an increase in *Firmicutes* [25].

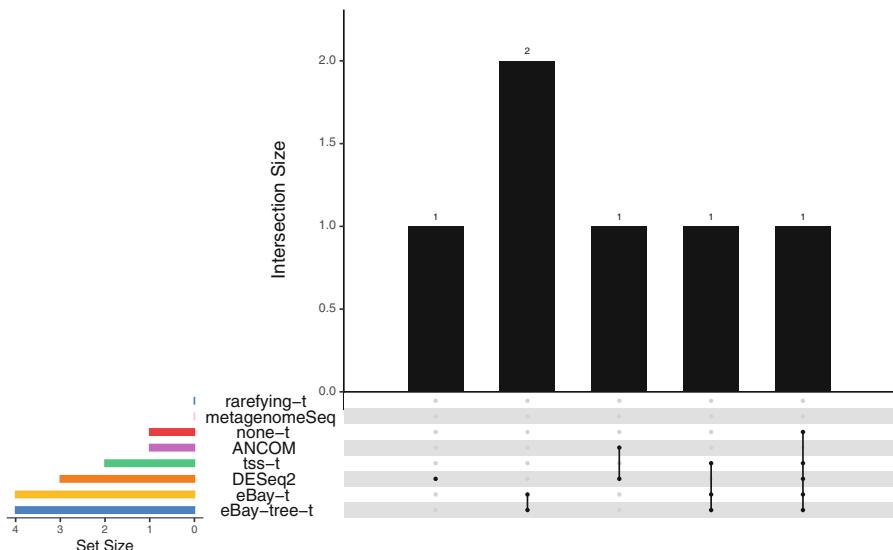


Fig. 2 Visualization of set intersections among differential abundance testing methods. ‘t’ stands for Welch’s t-test applied to data un-normalized (none) or normalized by rarefying, total sum scaling (tss), the empirical Bayes (eBay), or phylogeny-aware empirical Bayes (eBay-tree) method. Included in the figure are metagenomeSeq [39], ANCOM [32], and DESeq2 [29], all of which are the state-of-the-art methods for detecting differentially abundant OTUs. rarefying-t and metagenomeSeq failed to detect any OTUs, and eBay-t and eBay-tree-t identified the same set of four OTUs, of which two were unique, one was also found by tss-t, and one by none-t, tss-t, and DESeq2

Table 1 Taxonomic membership of four differentially abundant OTUs detected by Welch's t-test on empirical Bayes normalized data

OTU	Phylum	Class	Order	Family	Genus
25	<i>Bacteroidetes</i>	<i>Bacteroidetes</i>	<i>Bacteroidales</i>	<i>Bacteroidaceae</i>	<i>Bacteroides</i>
16,898	<i>Firmicutes</i>	<i>Clostridia</i>	<i>Clostridiales</i>	<i>Ruminococcaceae</i>	NA
13,906	<i>Firmicutes</i>	<i>Clostridia</i>	<i>Clostridiales</i>	<i>Lachnospiraceae</i>	NA
9409	NA	NA	NA	NA	NA

4 Regression with Compositional Predictors

In the previous sections, we discussed methods for analyzing multivariate abundance data. In some machine learning applications, microbiome data are used as predictors or inputs that have some influence on one or more responses or outputs [20]. For ease of exposition, this section is concerned with regression problems where the predictors are compositional data and the response is univariate. The compositional nature of relative abundances renders many standard regression methods inappropriate.

To remove the unit-sum constraint, [2] proposed a linear log-contrast model. Suppose that, in addition to $\mathbf{p}_i = (p_{i1}, \dots, p_{iK})^\top \in \mathbb{S}^{K-1}$ the vector of relative abundances, a response y_i is also observed on the i th sample. The model has the form

$$y_i = \sum_{j=1}^{K-1} \beta_j \log\left(\frac{p_{ij}}{p_{iK}}\right) + \epsilon_i, \quad (11)$$

for $i = 1, \dots, n$, where $\beta_1, \dots, \beta_{K-1}$ are the regression coefficients, and the errors ϵ_i are uncorrelated and have mean zero and constant variance. Note that, on the log scale, $\log(p_{ij}/p_{iK}) = \log(p_{ij}) - \log(p_{iK})$ is a linear contrast, hence the name linear log-contrast model. When K is small relative to n , we can simply fit this model by least squares.

4.1 Constrained Lasso and Log-Ratio Lasso

In metagenomic applications, however, the number of taxa, K , can be comparable to or larger than the number of observations, n , resulting in high-dimensional compositional data. In such settings, statistical learning is challenging for two reasons. The first is prediction accuracy: the traditional fitting procedures such as least squares and maximum likelihood often have low bias but large variance and hence suffer in prediction accuracy. The second reason is interpretability. With a large number of predictors, we would often like to determine a smaller subset that exhibits the strongest effects. As a result, regularized approaches that produce

sparse models often become the methods of choice. By shrinking the values of the regression coefficients to zero, these methods introduce some bias but reduce the variance of the predicted values and hence potentially lead to better generalization while at the same time allowing for easier interpretation of the coefficients. A popular example is the penalization of the squared loss in the classical linear model by the sum of absolute values of the coefficients, known as the lasso [56].

Clearly, the choice of the last component as baseline predictor in (11) is arbitrary. Let $\beta_K = \sum_{j=1}^{K-1} \beta_j$. We can rewrite (11) in a reference-free way as

$$y_i = \sum_{j=1}^K \beta_j \log(p_{ij}) + \epsilon_i, \quad \sum_{j=1}^K \beta_j = 0. \quad (12)$$

[27] proposed a lasso-based regularization method for fitting this model in high dimensions. They considered optimizing the constrained convex criterion

$$\underset{\beta_1, \dots, \beta_K, \sum_{j=1}^K \beta_j = 0}{\text{minimize}} \left[\sum_{i=1}^n \left\{ y_i - \sum_{j=1}^K \beta_j \log(p_{ij}) \right\}^2 + \lambda \sum_{j=1}^K |\beta_j| \right], \quad (13)$$

where λ is a tuning parameter. To solve this problem, they used the augmented Lagrangian method to develop a coordinate descent algorithm. For large enough values of λ , most coefficients will be zero.

The linear log-contrast model differs from the standard linear model in that it can be represented as a collection of log-contrasts, that is, log-ratios of two components, rather than just a vector of log-transformed components. As such, while the standard lasso is highly interpretable, a sparse solution from the constrained lasso does not necessarily correspond to an equally sparse collection of log-contrasts [4]. To see this, consider a toy example, in which $K = 8$ and the mean function has the form

$$E(Y | \mathbf{p}) = 2 \log \left(\frac{p_1}{p_3} \right) + \log \left(\frac{p_2}{p_4} \right).$$

Then, it is easy to see that

$$E(Y | \mathbf{p}) = (2 - c) \log \left(\frac{p_1}{p_3} \right) + c \log \left(\frac{p_1}{p_4} \right) + c \log \left(\frac{p_2}{p_3} \right) + (1 - c) \log \left(\frac{p_2}{p_4} \right),$$

for any constant c . Although the constrained lasso can identify the set $\{1, 2, 3, 4\}$, it cannot tell correctly which log-contrasts should be included in the model. In other words, the constrained lasso is not enough to enforce the desired sparsity in log-contrasts, and so it is quite different from the standard lasso (see Sect. 4.2 for another viewpoint).

To select a sparse collection of log-contrasts, [4] introduced the all-pairs log-ratio model

$$y_i = \sum_{1 \leq j < k \leq K}^K \theta_{jk} \log\left(\frac{p_{ij}}{p_{ik}}\right) + \epsilon_i, \quad (14)$$

where each of the θ_{jk} is a coefficient multiplying its associated log-ratio $\log(p_{ij}/p_{ik})$. This is an overparameterized version of the linear log-contrast model. Under the sparsity assumption that most of the coefficients θ_{jk} are zero, they proposed log-ratio lasso by minimizing

$$\sum_{i=1}^n \left\{ y_i - \sum_{1 \leq j < k \leq K}^K \theta_{jk} \log\left(\frac{p_{ij}}{p_{ik}}\right) \right\}^2 + \lambda \sum_{1 \leq j < k \leq K} |\theta_{jk}| \quad (15)$$

with respect to $\{\theta_{jk}, 1 \leq j < k \leq K\}$. Clearly, the notion of sparsity in log-ratio lasso is different from that in the constrained lasso.

The optimization problem is challenging because it is an optimization in $O(K^2)$ variables and requires explicit construction of the matrix of all-pairs log-ratios. Bates and Tibshirani [4] proposed a two-step procedure for finding a highly sparse solution. In the first screening step, the constrained lasso is fitted, and components with nonzero coefficients are identified. In the second pruning step, the selected components are used to enumerate all log-ratios, and a sparse regression such as forward stepwise regression is run. Screening makes sense, since the constrained lasso and log-ratio lasso are equivalent in terms of the fitted values. Pruning removes some ambiguity in log-contrast selection resulting from the solution to the constrained lasso.

Constrained lasso and log-ratio lasso

```
library(logratiolasso)
packageVersion("logratiolasso")

y <- metadata$bmi
x <- log(eBay.comps) # log of estimated compositions
centered_y <- y - mean(y)
centered_x <- scale(x, center=T, scale=F)

# constrained lasso
classo <- glmnet.constr(centered_x, centered_y)
set.seed(10)
cv_constr_lasso <- cv.glmnet.constr(classo,
                                         centered_x, centered_y)

# two-stage log-ratio lasso
set.seed(10)
```

```
cv_ts_lasso <- cv_two_stage(centered_x,
                               centered_y, k_max = 7)
```

4.2 Subcomposition Selection

Since a composition carries only relative information, subcompositions, which preserve ratio relationships, are fundamental objects of investigation in compositional data analysis. Indeed, subcompositional analysis has a long history, for example, in geology, and is a major theme of [1]. As such, in the regression setting, a natural counterpart of component selection should be subcomposition selection.

In order to describe this concept more precisely, we require some more notations. For a nonempty subset $\mathcal{S} \subseteq \{1, \dots, K\}$, define $p_j^{\mathcal{S}} = p_j / \sum_{k \in \mathcal{S}} p_k$, $j \in \mathcal{S}$ and $L(\mathcal{S}) = \sum_{j \in \mathcal{S}} \beta_j \log(p_j^{\mathcal{S}})$. We call $\{p_j^{\mathcal{S}}, j \in \mathcal{S}\}$, or \mathcal{S} for short, a subcomposition formed from the full composition $\{1, \dots, K\}$. If $\sum_{j \in \mathcal{S}} \beta_j = 0$, we call $L(\mathcal{S})$ a linear log-contrast of \mathcal{S} .

Under the linear log-contrast model (12), a subcomposition \mathcal{S} is said to be inactive if the expected response $E(Y | \mathbf{p})$ depends only on the subcomposition \mathcal{S}^c , the complement of \mathcal{S} ; \mathcal{S} is said to be active if the expected response $E(Y | \mathbf{p})$ depends on \mathcal{S} through $L(\mathcal{S})$. Note that when the cardinality of \mathcal{S} , denoted by $|\mathcal{S}|$, is 1, $\{p_j^{\mathcal{S}}, j \in \mathcal{S}\}$ reduces to {1} and $L(\mathcal{S}) = 0$; hence, \mathcal{S} is inactive.

Let $\mathcal{A} = \{j : \beta_j \neq 0\}$. Then, under model (12), $\sum_{j \in \mathcal{A}} \beta_j = 0$ and $E(Y | \mathbf{p}) = \sum_{j \in \mathcal{A}} \beta_j \log(p_j^{\mathcal{A}})$. By definition, \mathcal{A} is active as long as $|\mathcal{A}| > 1$. In other words, what the constrained lasso (or component selection in general) actually targets is a single subcomposition composed of selected components. This further sheds light on the difference between the linear log-contrast model and the standard linear model. Now consider the all-pairs log-ratio model (14). Since each log-ratio or log-contrast is a subcomposition of size 2, the log-ratio lasso performs subcomposition selection restricted to the set of simplest subcompositions.

In microbiome studies, an important objective is to identify groups of bacterial species present in an ecosystem that are predictive of a phenotype [20]. The constrained lasso provides only a rough solution, that is, a single group, for this purpose, a practical disadvantage. The log-ratio lasso, on the other hand, is overly restrictive at the other end of the spectrum. Furthermore, it has an identifiability issue. To illustrate this, consider another toy example, in which $K = 8$ and the mean function has the form

$$E(Y | \mathbf{p}) = \log\left(\frac{p_1}{p_3}\right) + \log\left(\frac{p_2}{p_4}\right).$$

Then, the set consisting of {5, 6, 7, 8} is inactive, the set consisting of {1, 2, 3, 4} is active, and the latter can be partitioned into two active subcompositions in two ways:

(i) {1, 3} and {2, 4} and (ii) {1, 4} and {2, 3}. There seems to be little to distinguish between (i) and (ii) in terms of goodness-of-fit.

Formally, assume that there is a partition of the full composition $\{1, \dots, K\}$ into $G + 1 \geq 2$ nonoverlapping subcompositions $\mathcal{S}_g, |\mathcal{S}_g| > 1, g = 1, \dots, G + 1$ so that

$$E(Y | \mathbf{p}) = \sum_{g=1}^G \sum_{j \in \mathcal{S}_g} \beta_{gj} \log(p_j^{\mathcal{S}_g}), \quad \sum_{j \in \mathcal{S}_g} \beta_{gj} = 0, \quad g = 1, \dots, G. \quad (16)$$

That is, the subcomposition \mathcal{S}_{G+1} is inactive, and the expected response depends on G subcompositions formed from a nonoverlapping partition of \mathcal{S}_{G+1}^c . To identify subcompositions that are predictive of the response, we need to infer G , the number of linear contrasts, and the corresponding coefficients within subcompositions.

The problem of subcomposition selection is challenging for two reasons. The first reason, as the toy example shows, is identifiability. The second is computation. The total number of all possible partitions of the full composition into subcompositions, which is the K th Bell number [43], is much larger than that of all possible subsets of components, and hence, it is computationally infeasible, even for a moderate K , to enumerate over all possible least squares regressions for identifying the best partition.

To address these challenges, [59] proposed a multiscale subcomposition selection method. Rather than searching through all possible solutions, they considered a setting where the relationships between the components can be represented as a tree, and proposed a tree-structured regularization method to select subcompositions at subtree levels. The motivation for their method is that microbial community changes can occur at different levels of granularity, and hence, finding microbial signatures at multiple granularities can both provide much insight into the underlying biology and improve prediction accuracy. We elaborate on this in the next section.

4.3 Phylogeny-Aware Subcomposition Selection

To avoid imposing the zero-sum constraint explicitly on the coefficients, define the centered log-ratio transformation $W_j = \log(p_j) - \sum_{k=1}^K \log(p_k)/K, j = 1, \dots, K$, and consider the following model:

$$Y = \sum_{j=1}^K \beta_j W_j + \epsilon. \quad (17)$$

Define $W_j^{\mathcal{S}} = \log(p_j^{\mathcal{S}}) - \sum_{k \in \mathcal{S}} \log(p_k^{\mathcal{S}})/|\mathcal{S}|, j \in \mathcal{S}$. Assume that

$$E(Y | \mathbf{W}) = \sum_{j=1}^K \beta_j W_j = \sum_{g=1}^G \sum_{j \in \mathcal{S}_g} \beta_{gj} W_j^{\mathcal{S}_g}, \quad (18)$$

where $\mathbf{W} = (W_1, \dots, W_K)^\top$. Note that for each $g \in \{1, \dots, G\}$, the coefficients β_{gj} in (18) are identifiable up to a common additive constant, due to the fact that $\sum_{j \in \mathcal{S}_g} W_j^{\mathcal{S}_g} = 0$. It is thus the relative, rather than absolute, value of β_{gj} that matters. This in turn implies that linear contrasts of β_{gj} , $j = 1, \dots, |\mathcal{S}_g|$, such as $\beta_{gj}^* = \beta_{gj} - \sum_{k \in \mathcal{S}_g} \beta_{gk}/|\mathcal{S}_g|$, $j = 1, \dots, |\mathcal{S}_g|$, are estimable.

Let $\beta_j^* = \beta_j - \sum_{k=1}^K \beta_k/K$, $j = 1, \dots, K$. Then,

$$E(Y | \mathbf{W}) = \sum_{j=1}^K \beta_j^* \log(p_j) = \sum_{g=1}^G \sum_{j \in \mathcal{S}_g} \beta_{gj}^* W_j^{\mathcal{S}_g}.$$

Therefore, model (12) with (16) and model (17) with (18) are equivalent.

Let v_0 denote the root node of \mathcal{T} . For each internal node $v \in \mathcal{I}$, let \mathcal{T}_v denote the subtree rooted at v . Clearly, for an internal node v near the bottom of \mathcal{T} , the components that correspond to the leaf nodes of \mathcal{T}_v are highly homogeneous, whereas for v near v_0 , the components associated with \mathcal{T}_v are relatively more heterogeneous. Consider now an arbitrary subcomposition. Because balanced trees are extremely rare, it is very likely that the components of this subcomposition are heterogeneous. To encourage the selection of homogeneous subcompositions, [59] proposed a tree-structured penalty function or regularizer.

Let \mathbf{e}_j be the K -dimensional vector whose j th element is 1 and other elements are 0, for $j = 1, \dots, K$. For each $v \in \mathcal{I} \cup \mathcal{L}$, denote by $\mathcal{L}_v \subseteq \{1, \dots, K\}$ the index set of the leaves of \mathcal{T}_v , and define $\mathbf{h}_v = \sum_{j \in \mathcal{L}_v} \mathbf{e}_j$. Note that \mathbf{h}_v represents a node-based group of components. Let $\boldsymbol{\beta}^* = (\beta_1^*, \dots, \beta_K^*)^\top$. The tree-guided penalty term is defined as

$$\begin{aligned} J^*(\boldsymbol{\beta}^*, \lambda_1, \lambda_2) &= \lambda_1 \sum_{v \in \mathcal{L}} |\mathbf{h}_v^\top \boldsymbol{\beta}^*| + \lambda_2 \sum_{v \in \mathcal{I}} |\mathbf{h}_v^\top \boldsymbol{\beta}^*| \\ &= \lambda_1 \sum_{j=1}^K |\beta_j^*| + \lambda_2 \sum_{v \in \mathcal{I} \setminus \{v_0\}} |\mathbf{h}_v^\top \boldsymbol{\beta}^*|, \end{aligned} \quad (19)$$

where λ_1 and λ_2 are regularization parameters. If $\mathbf{h}_v^\top \boldsymbol{\beta}^* = 0$ for some leaf node $v \in \mathcal{L}$, then the corresponding component is removed from the model. On the other hand, if $\mathbf{h}_v^\top \boldsymbol{\beta}^* = 0$ for an internal node $v \in \mathcal{I}$, then a partition occurs at v . As one moves from leaves to the root, the first time $\mathbf{h}_v^\top \boldsymbol{\beta}^* = 0$ happens at an internal node v , and this defines a subcomposition. Consequently, the first term in (19) is for component selection or elimination, while the second term is for subcomposition selection that induces homogeneity at the subtree level.

To select subcompositions and estimate parameters simultaneously, [59] considered the convex optimization problem:

$$\text{minimize}_{\boldsymbol{\beta}} \left[\sum_{i=1}^n \left(y_i - \sum_{j=1}^K \beta_j w_{ij} \right)^2 + \lambda_1 \sum_{j=1}^K |\beta_j^*| + \lambda_2 \sum_{v \in \mathcal{I} \setminus \{v_0\}} |\mathbf{h}_v^\top \boldsymbol{\beta}^*| \right], \quad (20)$$

where $\boldsymbol{\beta} = (\beta_1, \dots, \beta_K)^\top$. They called this method Tree-guided Automatic Subcomposition Selection Operator (TASSO). When $\lambda_2 = 0$, TASSO reduces to the constrained lasso.

Let \mathbf{e}_j^* and \mathbf{h}_v^* denote the centered versions of \mathbf{e}_j and \mathbf{h}_v , respectively. Then,

$$J^*(\boldsymbol{\beta}^*, \lambda_1, \lambda_2) = J(\boldsymbol{\beta}, \lambda_1, \lambda_2) = \lambda_1 \sum_{j=1}^K |\mathbf{e}_j^{*\top} \boldsymbol{\beta}| + \lambda_2 \sum_{v \in \mathcal{I} \setminus \{v_0\}} |\mathbf{h}_v^{*\top} \boldsymbol{\beta}|.$$

The criterion can then be written equivalently as

$$\text{minimize}_{\boldsymbol{\beta}} \left[\sum_{i=1}^n \left(y_i - \sum_{j=1}^K \beta_j w_{ij} \right)^2 + \lambda_1 \sum_{j=1}^K |\mathbf{e}_j^{*\top} \boldsymbol{\beta}| + \lambda_2 \sum_{v \in \mathcal{I} \setminus \{v_0\}} |\mathbf{h}_v^{*\top} \boldsymbol{\beta}| \right].$$

This is a generalized lasso problem and can be solved efficiently [57].

TASSO

```
# library(phyloMDA); packageVersion("phyloMDA")

fit_tasso <- TASSO(y, eBay.comps, tree)
fit_tasso

fit_classo <- TASSO(y, eBay.comps,
                      tree = NULL) # constrained lasso
fit_classo
```

4.4 Linear Regression and Variable Fusion

So far compositions are assumed to lie in a strictly positive simplex. The main reason is that we cannot take the logarithm of zero in log-contrasts or log-ratios.

In the absence of a one-to-one monotonic transformation between the real line and its non-negative subset, the problem of zeros might not be satisfactorily resolved, and solutions generally depend on the frequency and nature of the zeros.

There are three types of zeros: rounded zeros, sampling zeros, and structural zeros [33]. In the previous sections, we assume implicitly that all microbes are present in the microbial ecosystem and the zeros are the result of undersampling. However, in the presence of hundreds or thousands of bacterial species, these zeros can also represent components that are truly absent from the community, especially when the specimens are drawn from different environments [39]. This requires the treatment of compositions with zero components. Clearly, the strategy of replacing the structural zero by a small value and re-normalizing the data to have a unit sum is not appropriate.

We now consider the regression problem where the observed predictors are compositional and possibly with some zeros, that is, $p_{ik} \geq 0$, $\sum_{k=1}^K p_{ik} = 1$. Unfortunately, the presence of zeros causes log-ratio-based methods to fail in this case. To take into account the compositional nature, high dimensionality, and phylogeny of microbiome data, [61] introduced the concept of variable fusion and proposed a multiscale dimension reduction method. Instead of using the linear log-contrast model, they used the linear model

$$Y = \beta_0 + \sum_{j=1}^K \beta_j p_j + \epsilon. \quad (21)$$

This model has a similar flavor to model (17) in that the predictors are constrained to have a constant sum, $\sum_{k=1}^K p_k = 1$. It formally resembles standard analysis of variance, in which the dummy variables that code a multi-level categorical predictor sum up to one. It is thus easy to see that the coefficients β_j are identifiable only up to a common additive constant.

One can impose a constraint on the parameters to make them identifiable. Note that, for each $k \in \{1, \dots, K\}$,

$$E(Y) = \beta_0(k) + \sum_{j=1}^K \beta_j(k) p_j,$$

where $\beta_0(k) = \beta_0 + \beta_k$, and $\beta_j(k) = \beta_j - \beta_k$ reflects the difference between β_j and β_k . The constraint is then $\beta_k(k) = 0$.

The concept of variable fusion is motivated by an assumption that phylogenetically close taxa have similar associations with a host phenotype. Under this assumption, a good way to handle the dimensionality problem is to shrink some $\beta_{j_1} - \beta_{j_2}$ to zero. Since $\beta_{j_1}(k) - \beta_{j_2}(k) = \beta_{j_1} - \beta_{j_2}$, one can solve the pairwise fused lasso [45] problem

$$\text{minimize}_{\beta(k)} \left[\sum_{i=1}^n \left\{ y_i - \sum_{j=1}^K \beta_j(k) p_{ij} \right\}^2 + \lambda \sum_{1 \leq j_1 < j_2 \leq K} |\beta_{j_1}(k) - \beta_{j_2}(k)| \right], \quad (22)$$

where $\beta(k) = \{\beta_1(k), \dots, \beta_K(k)\}$, and we assume that the data have been mean centered, so that we can omit the intercept. The pairwise fused lasso is a generalization of the ordinary fused lasso [58], intended for situations in which variables have an ordering along which smoothness is expected. Since any ordering of OTUs is arbitrary, the ordinary fused lasso may be misleading.

The pairwise fused lasso treats all pairs of OTUs equally and thus fails to exploit the phylogeny of the OTUs. One can use a weighted penalty that smoothes the coefficients of two OTUs j_1 and j_2 based on their closeness $d_{j_1 j_2}$ on the tree:

$$\lambda \sum_{1 \leq j_1 < j_2 \leq K} \omega_{j_1 j_2} |\beta_{j_1}(k) - \beta_{j_2}(k)|,$$

where $\omega_{j_1 j_2} = d_{j_1 j_2}^\gamma$ for some $\gamma \leq 0$.

Variable fusion is crucial in high dimensions for improved predictive performance. It is immune to zeros and is operationally adapted to the compositional nature of the data. For increased interpretability, [61] proposed tree-guided variable fusion to harness a predictive microbial signature made of a set of multi-level taxa. They constructed two weighted fused lasso penalties that encode the tree topology.

For simplicity, assume that the phylogenetic tree $\mathcal{T} = (\mathcal{L}, \mathcal{I})$ is binary. For each internal node $v \in \mathcal{I}$, let c_{v1} and c_{v2} be the two child nodes of v . The first penalty is defined by

$$\lambda \sum_{v \in \mathcal{I}} \omega_{c_{v1} c_{v2}} |\mathbf{s}_v^\top \boldsymbol{\beta}(s)|, \quad (23)$$

where $\mathbf{s}_v \in \mathbb{R}^K$ is an indicator vector with j th entry $1/|\mathcal{L}_{c_{v1}}|$ if $j \in \mathcal{L}_{c_{v1}}$, $-1/|\mathcal{L}_{c_{v2}}|$ if $j \in \mathcal{L}_{c_{v2}}$, and 0 otherwise. In other words, the two child nodes c_{v1} and c_{v2} each take a proportion of the weight $\omega_{c_{v1} c_{v2}}$ of the parent node $v \in \mathcal{I}$, relative to the sizes of their subtrees.

The second penalty is defined in a bottom-up recursive manner by first computing the penalty terms for all internal nodes with size 2 subtrees, then all with size 3 subtrees if any exist, and so on. Specifically, define \mathcal{A} to be the level set such that, for each $l \in \mathcal{A}$, there is $v \in \mathcal{I}$ such that $l = |\mathcal{L}_v|$. Let l_h denote the h th smallest element of \mathcal{A} . For each $v \in \mathcal{L} = \{1, \dots, K\}$, let \mathbf{e}_v be the K -dimensional vector whose v th element is 1 and other elements are 0, and for each $v \in \mathcal{I}$, initialize \mathbf{e}_v to be the K -vector of zeros. For $h = 1, \dots, |\mathcal{A}|$, recursively set $\mathbf{t}_v = \mathbf{e}_{c_{v1}} - \mathbf{e}_{c_{v2}}$ and update $\mathbf{e}_v = (\mathbf{e}_{c_{v1}} + \mathbf{e}_{c_{v2}})/2$, for all $v \in \mathcal{I}$ such that $|\mathcal{L}_v| = l_h$. The second penalty is defined by

$$\lambda \sum_{v \in \mathcal{I}} \omega_{c_{v1}c_{v2}} |\mathbf{t}_v^\top \boldsymbol{\beta}(k)|. \quad (24)$$

Consider then the tree-guided fused lasso problem

$$\text{minimize}_{\boldsymbol{\beta}(k)} \left[\sum_{i=1}^n \left\{ y_i - \sum_{j=1}^K \beta_j(k) p_{ij} \right\}^2 + \lambda \sum_{1 \leq j_1 < j_2 \leq K} \omega_{c_{j_1}c_{j_2}} |\mathbf{c}_{j_1}^\top \boldsymbol{\beta}(k)| \right], \quad (25)$$

where $\mathbf{c}_v = \mathbf{s}_v$ or $\mathbf{c}_v = \mathbf{t}_v$. Note that, for each $v \in \mathcal{I}$, $\mathbf{c}_v^\top \boldsymbol{\beta}(k) = \mathbf{c}_v^\top \boldsymbol{\beta}$ is a linear contrast of the coefficients β_j , hence the name tree-guided fused lasso. Again, the optimization problem (25) has a generalized lasso formulation.

Since each internal node represents the abundance of a taxonomic lineage, by incorporating the tree information node by node, the estimated microbial signature from the tree-guided fused lasso tends to be composed of a few taxonomic units at different depths. Formally, the variables indexed by \mathcal{L}_u for $u \in \mathcal{I}$ are fused together, defining a new variable indexed by u , if $\mathbf{c}_v^\top \boldsymbol{\beta} = 0$ for all the internal nodes v of the subtree rooted at u .

Tree-guided fused lasso

```
# library(phyloMDA); packageVersion("phyloMDA")

fit_tflasso1 <- TreeFusedlasso(y, eBay.comps, tree)
fit_tflasso1

fit_tflasso2 <- TreeFusedlasso(y, eBay.comps, tree,
                                type = 2)
fit_tflasso2
```

5 Additional References

In addition to the Dirichlet prior, useful distributions for the MN probabilities include the Dirichlet mixture prior [18] and Aitchison's logistic normal (ALN) distribution [6]. In particular, the compound distribution combining ALN with MN accommodates a much richer dependence structure among bacterial counts than the DM distribution [63, 67]. Note that the MN distribution and its extensions condition on the total count in a sample. An alternative strategy is to analyze the multivariate counts unconditionally using, for example, Poisson graphical models

[5, 62], the multivariate Poisson-lognormal distribution [3], latent variable models [48], or copula models [49]; see [19] for a good review.

Generally speaking, normalization is a process that transforms the data from different samples to enable meaningful comparison. The effect of microbiome data normalization can be quantified using results from downstream analyses, such as ordination analysis and differential abundance testing [37, 55, 65]. Besides rarefying and TSS, there are a number of other normalization methods, including scaling normalization from the RNA-seq field [23, 39] and log-ratio transformation that accounts for the compositional nature of sequencing data [17, 32]. TSS is a naive scaling method, and when log-ratio transformed, raw proportions from TSS are operationally equivalent to the raw counts. Microbiome data are compositional and should be treated as compositions. In this sense, scaling normalization is unnecessary. However, zeros cannot be log-transformed, and when the normalized data from scaling have zeros, they will lead to difficulties in downstream analyses. The (empirical) Bayesian formulation in Sect. 3 addresses this issue. Using a Poisson-multinomial model for read counts, [7] proposed a regularized maximum likelihood approach to estimate the composition matrix.

For increased interpretability and the inclusion of taxonomic information, [46] extended constrained lasso to include multiple linear constraints. Lu et al. [31] further developed generalized linear models with linear constraints for microbiome compositional data. To improve prediction accuracy, [38, 52] proposed phylogenetic approaches to microbial community classification, and [15] introduced phylogenetic convolutional neural networks in metagenomics.

6 Discussion

Microbiome count data contain a high proportion of zeros. In Sects. 2 and 3, we assumed implicitly that all microbes are present in the samples and the zeros are the result of undersampling. However, not all zeros are the same, and sequence count data can exhibit zero inflation, especially when specimens are drawn from different environments. How to model multivariate count data that accounts for over-dispersion, zero inflation, and the phylogeny is an important research topic. The zero-inflated generalized Dirichlet-multinomial model [54] provides a possible solution to this problem.

Due to contamination, extraction, amplification, sequencing, and other technical biases, the interpretation of microbiome data is challenging, since the comparison of taxon relative abundances in the specimen is not equivalent to the comparison of taxon true abundances in the ecosystem from which the specimen was obtained. As in Sect. 3, it is reasonable to estimate the relative abundance of a taxon in the ecosystem using its relative abundance in the specimen. Whether or not we can use the specimen-level abundance data to draw inferences about taxon abundances at the ecosystem level is interesting but is less developed in the literature [13, 32].

In this chapter, we assumed that the phylogenetic tree is known a priori. In practice, the phylogeny is inferred from molecular sequences [41, 51], and so it is important to incorporate uncertainty in phylogenetic inference into downstream analyses. Another problem with tree is rooting. The phylogeny can be rooted using the outgroup or midpoint rooting method. However, since rooting is not part of tree inference, rooting error is in addition to tree-estimation error. Robust methods for integrating unrooted phylogenies in data analysis are highly demanding.

Acknowledgments We would like to thank Tiantian Liu and Chao Zhou for contributions on the R package *phyloMDA*. This research was supported in part by the National Natural Science Foundation of China (11971017), National Key R&D Program of China (2018YFC0910500), Shanghai Municipal Science and Technology Major Project (2017SHZDZX01), SJTU Trans-med Awards Research Young Faculty Grant (YG2019QNA26, YG2019QNA37), and Neil Shen's SJTU Medical Research Fund.

References

1. Aitchison, J.: The Statistical Analysis of Compositional Data. Springer, New York (1986)
2. Aitchison, J., Bacon-Shone, J.: Log contrast models for experiments with mixtures. *Biometrika* **71**(2), 323–330 (1984)
3. Aitchison, J., Ho, C.H.: The multivariate Poisson-log normal distribution. *Biometrika* **76**(4), 643–653 (1989)
4. Bates, S., Tibshirani, R.: Log-ratio lasso: scalable, sparse estimation for log-ratio models. *Biometrics* **75**(2), 613–624 (2019)
5. Besag, J.: Spatial interaction and the statistical analysis of lattice systems. *J. R. Stat. Soc. Ser. B* **36**(2), 192–236 (1974)
6. Billheimer, D., Guttorp, P., Fagan, W.F.: Statistical interpretation of species composition. *J. Am. Stat. Assoc.* **96**(456), 1205–1214 (2001)
7. Cao, Y., Zhang, A., Li, H.: Multisample estimation of bacterial composition matrices in metagenomics data. *Biometrika* **107**(1), 75–92 (2020)
8. Caporaso, J.G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F.D., Costello, E.K., et al.: QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods* **7**(5), 335–336 (2010)
9. Chen, J., Li, H.: Variable selection for sparse Dirichlet-multinomial regression with an application to microbiome data analysis. *Ann. Appl. Stat.* **7**(1), 418–442 (2013)
10. Connor, R.J., Mosimann, J.E.: Concepts of independence for proportions with a generalization of the Dirichlet distribution. *J. Am. Stat. Assoc.* **64**(325), 194–206 (1969)
11. Dennis, S.Y.: On the hyper-Dirichlet type 1 and hyper-Liouville distributions. *Commun. Stat. Theory Methods* **20**(12), 4069–4081 (1991)
12. Egozcue, J.J., Pawlowsky-Glahn, V., Mateu-Figueras, G., Barcelo-Vidal, C.: Isometric logratio transformations for compositional data analysis. *Math. Geol.* **35**(3), 279–300 (2003)
13. Fang, H., Huang, C., Zhao, H., Deng, M.: CCLasso: correlation inference for compositional data through lasso. *Bioinformatics* **31**(19), 3172–3180 (2015)
14. Fernandes, A.D., Reid, J.N., Macklaim, J.M., McMurrrough, T.A., Edgell, D.R., Gloor, G.B.: Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16S rRNA gene sequencing and selective growth experiments by compositional data analysis. *Microbiome* **2**(1), 15 (2014)
15. Fioravanti, D., Giarratano, Y., Maggio, V., Agostinelli, C., Chierici, M., Jurman, G., et al.: Phylogenetic convolutional neural networks in metagenomics. *BMC Bioinf.* **19**, 49 (2018)

16. Friedman, J., Alm, E.J.: Inferring correlation networks from genomic survey data. *PLoS Comput. Biol.* **8**(9), e1002687 (2012)
17. Gloor, G.B., Macklaim, J.M., Pawlowsky-Glahn, V., Egozcue, J.J.: Microbiome datasets are compositional: and this is not optional. *Front. Microbiol.* **8**, 1–6 (2017)
18. Holmes, I., Harris, K., Quince, C.: Dirichlet multinomial mixtures: generative models for microbial metagenomics. *PLoS One* **7**(2), e30126 (2012)
19. Inouye, D.I., Yang, E., Allen, G.I., Ravikumar, P.: A review of multivariate distributions for count data derived from the Poisson distribution. *Wiley Interdiscipl. Rev. Comput. Stat.* **9**, e1398 (2017)
20. Knights, D., Parfrey, L.W., Zaneveld, J., Lozupone, C., Knight, R.: Human-associated microbial signatures: examining their predictive value. *Cell Host Microbe* **10**(4), 292–296 (2011)
21. Knight, R., Vrbanac, A., Taylor, B.C., Aksenov, A., Callewaert, C., Debelius, J., et al.: Best practices for analysing microbiomes. *Nat. Rev. Microbiol.* **16**(7), 410–422 (2018)
22. Kuczynski, J., Lauber, C.L., Walters, W.A., Parfrey, L.W., Clemente, J.C., Gevers, D., et al.: Experimental and analytical tools for studying the human microbiome. *Nat. Rev. Genet.* **13**(1), 47–58 (2012)
23. Kumar, M.S., Slud, E.V., Okrah, K., Hicks, S.C., Hannenhalli, S., Bravo, H.C.: Analysis and correction of compositional bias in sparse sequencing count data. *BMC Genom.* **19**(1), 1–23 (2018)
24. La Rosa, P.S., Brooks, J.P., Deych, E., Boone, E.L., Edwards, D.J., Wang, Q., et al.: Hypothesis testing and power calculations for taxonomic-based human microbiome data. *PLoS One* **7**(12), e52078 (2012)
25. Ley, R.E.: Obesity and the human microbiome. *Curr. Opin. Gastroenterol.* **26**(1), 5–11 (2010)
26. Li, H.: Microbiome, metagenomics, and high-dimensional compositional data analysis. *Ann. Rev. Stat. Appl.* **2**, 73–94 (2015)
27. Lin, W., Shi, P., Feng, R., Li, H.: Variable selection in regression with compositional covariates. *Biometrika* **104**(4), 785–797 (2014)
28. Liu, T., Zhao, H., Wang, T.: An empirical Bayes approach to normalization and differential abundance testing for microbiome data. *BMC Bioinformatics* **21**, 225 (2020)
29. Love, M.I., Huber, W., Anders, S.: Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**(12), 550–550 (2014)
30. Lozupone, C., Lladser, M.E., Knights, D., Stombaugh, J., Knight, R.: UniFrac: an effective distance metric for microbial community comparison. *ISME J.* **5**(2), 169–172 (2011)
31. Lu, J., Shi, P., Li, H.: Generalized linear models with linear constraints for microbiome compositional data. *Biometrics* **75**(1), 235–244 (2019)
32. Mandal, S., Treuren, W.V., White, R., Eggesbø, M.Å., Knight, R., Peddada, S.D.: Analysis of composition of microbiomes: a novel method for studying microbial composition. *Microb. Ecol. Health Dis.* **26**(1), 27663–27663 (2015)
33. Martin Fernandez, J.A., Palarea-Albaladejo, J., Olea, R.A.: Dealing with zeros. In: Pawlowsky-Glahn, V., Buccianti, A. (eds.) *Compositional Data Analysis: Theory and Applications*, chap. 4, pp. 47–82. Wiley, London (2011)
34. Martin-Fernandez, J.A., Hron, K., Templ, M., Filzmoser, P., Palarea-Albaladejo, J.: Bayesian-multiplicative treatment of count zeros in compositional data sets. *Stat. Modell.* **15**(2), 134–158 (2015)
35. Martiny, J.B., Jones, S.E., Lennon, J.T., Martiny, A.C.: Microbiomes in light of traits: a phylogenetic perspective. *Science* **350**(6261), aac9323 (2015)
36. McMurdie, P.J., Holmes, S.: phyloseq: an R package for reproducible interactive analysis and graphics of microbiome census data. *PLoS One* **8**(4), e61217 (2013)
37. McMurdie, P.J., Holmes, S.: Waste not, want not: why rarefying microbiome data is inadmissible. *PLoS Comput. Biol.* **10**(4), e1003531 (2014)
38. Ning, J., Beiko, R.G.: Phylogenetic approaches to microbial community classification. *Microbiome* **3**(1), 47–54 (2015)
39. Paulson, J.N., Stine, O.C., Bravo, H.C., Pop, M.: Differential abundance analysis for microbial marker-gene surveys. *Nat. Methods* **10**(12), 1200–1202 (2013)

40. Pawlowsky-Glahn, V., Egozcue, J.J., Tolosana-Delgado, R.: *Modeling and Analysis of Compositional Data*. Wiley, London (2015)
41. Price, M.N., Dehal, P.S., Arkin, A.P.: FastTree 2—approximately maximum-likelihood trees for large alignments. *PLoS One* **5**(3), e9490 (2010)
42. Proctor, L.M., Creasy, H.H., Fettweis, J.M., Lloyd-Price, J., Mahurkar, A., Zhou, W.Y., et al.: The integrative human microbiome project. *Nature* **569**(7758), 641–648 (2019)
43. Rota, G.: The number of partitions of a set. *Am. Math. Month.* **71**(5), 498–504 (1964)
44. Rothschild, D., Weissbrod, O., Barkan, E., Kurilshikov, A., Korem, T., Zeevi, D., et al.: Environment dominates over host genetics in shaping human gut microbiota. *Nature* **555**(7695), 210–215 (2018)
45. She, Y.: Sparse regression with exact clustering. *Electron. J. Stat.* **4**, 1055–1096 (2010)
46. Shi, P., Zhang, A., Li, H.: Regression analysis for microbiome compositional data. *Ann. Appl. Stat.* **10**(2), 1019–1040 (2016)
47. Silverman, J.D., Washburne, A.D., Mukherjee, S., David, L.A.: A phylogenetic transform enhances analysis of compositional microbiota data. *Elife* **6**, e21887 (2017)
48. Skrondal, A., Rabe-Hesketh, S.: *Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models*. Chapman & Hall/CRC, Boca Raton (2004)
49. Song, P.: Multivariate dispersion models generated from Gaussian copula. *Scand. J. Stat.* **27**(2), 305–320 (2000)
50. Song, Y., Zhao, H., Wang, T.: An adaptive independence test for microbiome community data. *Biometrics* **76**(2), 414–426 (2020)
51. Stamatakis, A.: RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **30**(9), 1312–1313 (2014)
52. Tanaseichuk, O., Borneman, J., Jiang, T.: Phylogeny-based classification of microbial communities. *Bioinformatics* **30**(4), 449–456 (2014)
53. Tang, Y., Ma, L., Nicolae, D.L.: A phylogenetic scan test on a Dirichlet-tree multinomial model for microbiome data. *Ann. Appl. Stat.* **12**(1), 1–26 (2018)
54. Tang, Z.-Z., Chen, G.: Zero-inflated generalized Dirichlet multinomial regression model for microbiome compositional data analysis. *Biostatistics* **20**(4), 698–713 (2019)
55. Thorsen, J., Brejnrod, A.D., Mortensen, M.S., Rasmussen, M.A., Stokholm, J., Al-Soud, W.A., et al.: Large-scale benchmarking reveals false discoveries and count transformation sensitivity in 16S rRNA gene amplicon data analysis methods used in microbiome studies. *Microbiome* **4**(1), 62 (2016)
56. Tibshirani, R.: Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B* **58**(1), 267–288 (1996)
57. Tibshirani, R.J., Taylor, J.: The solution path of the generalized lasso. *Ann. Stat.* **39**(3), 1335–1371 (2011)
58. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. *J. R. Stat. Soc. Ser. B* **67**(1), 91–108 (2005)
59. Wang, T., Zhao, H.: Structured subcomposition selection in regression and its application to microbiome data analysis. *Ann. Appl. Stat.* **11**(2), 771–791 (2017)
60. Wang, T., Zhao, H.: A Dirichlet-tree multinomial regression model for associating dietary nutrients with gut microorganisms. *Biometrics* **73**(3), 792–801 (2017)
61. Wang, T., Zhao, H.: Constructing predictive microbial signatures at multiple taxonomic levels. *J. Am. Stat. Assoc.* **112**(519), 1022–1031 (2017)
62. Wang, T.: Graph-assisted inverse regression for count data and its application to sequencing data. *J. Comput. Graph. Stat.* **29**(3), 444–454 (2020)
63. Wang, T., Yang, C., Zhao, H.: Prediction analysis for microbiome sequencing data. *Biometrics* **75**(3), 875–884 (2019)
64. Washburne, A.D., Morton, J.T., Sanders, J., McDonald, D., Zhu, Q., Oliverio, A.M., et al.: Methods for phylogenetic analysis of microbiome data. *Nature Microbiol.* **3**(6), 652–661 (2018)

65. Weiss, S., Xu, Z., Peddada, S., Amir, A., Bittinger, K., Gonzalez, A., et al.: Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome* **5**(1), 27 (2017)
66. Wu, G.D., Chen, J., Hoffmann, C., Bittinger, K., Chen, Y.-Y., Keilbaugh, S.A., et al.: Linking long-term dietary patterns with gut microbial enterotypes. *Science* **334**(6052), 105–108 (2011)
67. Xia, F., Chen, J., Fung, W.K., Li, H.: A logistic normal multinomial regression model for microbiome compositional data analysis. *Biometrics* **69**(4), 1053–1063 (2013)
68. Zhang, Y., Zhou, H., Zhou, J., Sun, W.: Regression models for multivariate count data. *J. Comput. Graph. Stat.* **26**(1), 1–13 (2017)

A Log-Linear Model for Inference on Bias in Microbiome Studies



Ni Zhao and Glen A. Satten

1 Introduction

Microbiome studies are known to be biased, and almost all steps in the experimental, sequencing, and bioinformatic analysis pipeline are potential culprits. For example, DNA extraction protocols differ in their ability to lyse certain bacterial cells (e.g., Gram-positive cells) and therefore may preferentially obtain DNA from some taxa over others [4, 10]. PCR bias can be introduced by differences in the GC content of the microbiome sequences. Different bacterial sequences may bind differentially to primers, preferentially amplifying some taxa compared to others [5, 6, 11, 14]. Commonly used sequencing platforms (e.g., MiSeq and HiSeq) also differ in their ability to correctly read DNA when GC content is high [12]. Every step in the bioinformatic processing pipeline, including read filtering, trimming, deduplication, read mapping, choice of amplicon clustering method, and choice of reference database, can also produce bias [13]. Differences in gene copy numbers also contribute to bias [7]. Because these biases are protocol- and taxon-dependent, microbiome data generated from different protocols are quantitatively incomparable, and analyses that do not account for bias may lead to spurious conclusions. However, modeling every possible source of bias factors is a daunting process.

Recently, McLaren, Willis and Callahan (MWC) [9] proposed a simple model for the bias generation process in microbiome sequencing studies. MWC demonstrated that their model fits mock (or model) community data where true taxa relative

N. Zhao

Department of Biostatistics, Johns Hopkins University, Baltimore, MD, USA
e-mail: nzhao10@jhu.edu

G. A. Satten (✉)

Department of Gynecology and Obstetrics, Emory University, Atlanta, GA, USA
e-mail: gsatten@emory.edu

abundances are known. In their model, the observed relative abundance of each taxon (e.g., operational taxonomic unit (OTU) or amplicon sequence variant (ASV)) is a product of the true taxon prevalence and a taxon-specific *bias factor*, normalized over all taxa observed in the sample. Each bias factor represents the accumulation of multiplicative biases over all the steps in the experimental pipeline, so that the bias can be described by a single factor for each taxon. In particular, there is no need to consider taxon–taxon interactions, i.e., the effect that one taxon in a sample might have on the bias factor of a second taxon. However, the *bias* of any taxon at the relative abundance level can still be impacted by other taxa when converting observed counts into relative abundances, as the normalization factor depends on true prevalence and bias factors of *all* taxa in the sample.

The MWC model is important for at least two reasons. First, the MWC model assertion of no taxon–taxon interaction, if true, provides a simple way to discuss biases one taxon at a time, rather than at the community or sample level, making calibration of bias easier in studies involving large microbiome communities. One goal of this chapter is to develop methods to test this hypothesis. Second, the MWC model, via its multiplicative structure of bias factors, gives a simple way to describe the relationship between protocol and sample bias in a microbiome experiment, acknowledging that the bias factors are not properties of the microbes alone but depend heavily on experimental details such as the extraction protocol and PCR parameters [4–6, 10, 11, 14]. Thus, a taxon may have a low bias factor using one DNA extraction protocol but may have a high bias factor using a different extraction protocol.

Because the MWC model relates observed and true prevalences, it is necessary to know the true prevalence of a taxon before its bias factor can be calculated. As a result, calculation of bias factors is presumably restricted to model communities in which the true prevalences are known by construction. This simplifies the analysis in many ways. We know which taxa are present in each sample, so the zeros in the count table corresponding to bacteria that were not included in a model communities sample correspond to “structural missingness.” However, complex patterns of structurally missing data can also lead to complications in parameter estimation and statistical inference on bias factors. For example, permutation or bootstrap-based inference is more difficult because residuals of two samples with different composition are not exchangeable.

The first goal of this chapter is to develop a statistical model that generalizes the MWC model to allow inference on complex questions about bias factors. MWC focuses primarily on graphical demonstration of their model and offers limited capacity for statistical inference. In particular, MWC does not propose a statistical model for model communities data, relying instead on geometric arguments to estimate bias parameters in only the simplest situations. In addition, MWC does not consider covariates that could affect bias factors (such as plate effects or variations in extraction protocol). Here, we generalize the MWC model to include such covariates, and show the resulting model is a log-linear compositional model. We develop a novel estimation procedure for bias factors and propose permutation-

based inference for complex hypotheses involving bias factors across taxa and protocols.

The second goal of this chapter is to use the framework we develop to test the fundamental assumption in the MWC model that the presence of one bacteria does not affect the bias factors of any other bacteria in the sample. It should be clear that this hypothesis requires samples that have a variable number of bacteria in each sample. Published mock community data of Brooks et al. [2] have this unusual feature. The structural missingness implied by this requirement motivates much of our methods' development. The Brooks' data also features three types of samples (cells, DNA, and PCR products); further, each type of sample was run on two plates. Thus, we also use these data to address simpler questions like whether there is evidence of differential bias across sample types or across plates.

The rest of this chapter is organized as follows. In Sect. 2.1, we set the stage by describing the Brooks' data and the scientific questions it raises. In Sects. 2.2 and 2.3, we describe our methods of analysis. Section 3 presents the performance of our model over a variety of realistic simulation scenarios. We describe the results of our analysis of the Brooks' data in Sect. 4.2. Some technical details are relegated to an Appendix. A final section gives concluding remarks.

2 Methods

2.1 The Brooks' Data

Brooks et al. [2] conducted a fairly large-scale model communities study using seven bacteria commonly found in vaginal samples, viz., *L. crispatus*, *L. iners*, *G. vaginalis*, *A. vaginae*, *P. bivia*, *S. amnii*, and group B streptococcus (GBS). Each mock sample consists of an even mixture of one to seven bacterial taxa; in total, there were 58 unique combinations of bacteria. Of the 240 total samples, 27 samples have only a single bacterium present, 75 have two bacteria, 129 have three bacteria, 4 have three bacteria, and 6 have all seven bacteria. Here, we ignore the samples having only one bacterium as they are uninformative in both the MWC model and our approach. In addition, three versions of every sample were created: one from an even mixture of cells, one from an even mixture of extracted DNA, and one from an even mixture of PCR products. The samples were processed on six distinct plates with two plates for the cell samples, two for the DNA samples, and two for the PCR product samples. Here, a *plate* refers to a set of samples that were processed and sequenced together. Because of the fundamental difference in sample types across plates, it is of interest to ask if the biases differ both within and between plates. The study was well-balanced across plates with respect to the numbers of taxa per sample ($p = 0.974$), and since the smallest nonzero taxon relative abundance is $1/7$, the nominally significant differences in library size across plates ($p = 0.002$) have little effect. Table 1 provides a description of the samples in this dataset.

Table 1 Descriptive statistics for the Brooks' data

Sample types	Cell		DNA		PCR		<i>p</i> -values
Plates	1	2	3	4	5	6	
# of samples	40	40	40	40	40	40	
Library size (mean)	16.2K	17.2K	12.8K	14.4K	15.3K	15.2 K	0.002
Number of taxa present (%)							0.974
1	3 (7.5)	6 (15.0)	3 (7.5)	6 (15.0)	3 (7.5)	6 (15.0)	
2	15 (37.5)	10 (25.0)	15 (37.5)	10 (25.0)	15 (37.5)	10 (25.0)	
3	20 (50.0)	23 (57.5)	20 (50.0)	23 (57.5)	20 (50.0)	23 (57.5)	
4	1 (2.5)	0 (0.0)	1 (2.5)	0 (0.0)	1 (2.5)	0 (0.0)	
7	1 (2.5)	1 (2.5)	1 (2.5)	1 (2.5)	1 (2.5)	1 (2.5)	

Although the pattern of structural missingness makes inference more difficult, this design allows us to address the fundamental assumption of the MWC model, by asking whether the presence of one bacterium (say, *L. iners*) affects the bias factors of the other bacteria, and whether this interaction effect is consistent across sample types and plates. In this chapter, we generally denote the potential effect of one taxon on the bias factor of other taxa as the “interaction” effect, as compared to the “main” effect that describes the impact of external factors such as sequencing protocols, sample type, and plate effect.

2.2 Setup and Estimation

We assume that the experimental data can be summarized in a count table (i.e., OTU or ASV table) where the N rows correspond to samples and the J columns to taxa (i.e., OTUs or ASVs) that occur in at least one sample. Let \tilde{p}_{ij} denote the observed relative abundance (or prevalence) of the j th taxon in the i th sample, calculated as the observed counts of the j th taxon divided by the library size of the sample, $i \in (1, \dots, N)$, $j \in (1, \dots, J)$. We let p_{ij} denote the true relative abundance of the j th taxon in the i th sample, assumed known. We further let $\Delta_{ij} = 1$ if the j th taxon is known to be present in the i th sample, and take $\Delta_{ij} = 0$ otherwise. Since we are considering model communities data, we restrict the analysis to include all taxa known to be present in a sample, so that we are assured that $\sum_{j=1}^J \tilde{p}_{ij} \Delta_{ij} = \sum_{j=1}^J p_{ij} \Delta_{ij} = 1$ for each sample. Here, we assume that $\tilde{p}_{ij} > 0$ whenever $\Delta_{ij} > 0$. This is a reasonable assumption in mock community data with reasonable sequencing depth, in which the number of taxa is small, and all taxa have at least moderate relative abundances. In the unlikely case that $\tilde{p}_{ij} = 0$ while $\Delta_{ij} > 0$, the taxon can be taken to be absent from that sample and the p_{ij} s adjusted accordingly.

The MWC model (equation 4 in [9]) asserts that

$$E(\tilde{p}_{ij}) = \frac{p_{ij} \exp(\beta_j)}{\sum_{j'=1}^J p_{ij'} \exp(\beta_{j'})}, \quad (1)$$

where $\exp(\beta_j)$ is the bias factor for taxon j . This model is equivalent to the following:

$$\ln E(\tilde{p}_{ij}) = \ln p_{ij} + \beta_j + \alpha_i, \text{ for } \Delta_{ij} > 0,$$

in which α_i is a sample-specific normalization. Equation (1) could be used as the mean structure for a cell count model, assuming an underlying distribution for the counts for each taxon (such as the Dirichlet-multinomial mixture model). However, these models can be hard to fit, and the parametric assumptions underlying them are unattractive. MWC also proposes a model (equation 10 in [9]) with nearly the same mean structure

$$\ln \tilde{p}_{ij} = \ln p_{ij} + \beta_j + \alpha_i + \epsilon_{ij}, \text{ for } \Delta_{ij} > 0, \quad (2)$$

in which ϵ_{ij} is an error term. MWC did not fully differentiate between these two models, which are, strictly, not consistent with each other. Nevertheless, when the sequencing depth is modest or high and the abundance \tilde{p}_{ij} is not too low, as is the case in mock communities samples, the mean structure of (2) gives a close approximation to the mean structure in (1). Although we did not implement this in most of our analyses, a Haldane-like correction (corresponding to adding 1 count to both the numerator and denominator when calculating the observed relative abundance) can be applied so that $E(\ln \tilde{p}_{ij})$ better approximates $\ln E(\tilde{p}_{ij})$ if small read counts are expected. In our simulation studies in Sect. 3, we deliberately simulated data using a Dirichlet-multinomial mixture model having the MWC mean structure (1) instead of log-linear model data generated using (2), to demonstrate that our proposed model works well for either of these underlying data generation processes.

Model (2) has the advantage that it allows us to use the simple machinery of least squares for parameter estimation and inference. No parametric assumption on the distribution of ϵ_{ij} is needed except that $E(\epsilon_{ij}) = 0$ and the existence of the second moment. To allow for sample-level covariates, let X be a $N \times M$ design matrix with i th row corresponding to covariates for the i th sample. Examples of covariates in X include the experimental procedures in the sequencing pipeline such as the DNA extraction method, the primers used and the sequencing machine, or more generally, plates (batches) in which the samples were processed. M is the total number of covariates we want to investigate. We adopt notation in which Q_k . ($Q_{\cdot k}$) is the row (column) vector corresponding to the k th row (column) of matrix Q . Through X , we can investigate important biological questions such as the effect of DNA extraction method on the bias factors and the taxon–taxon interactions. Let β be a $M \times J$ matrix of parameters. With this, we generalize the MWC model to

$$\ln \tilde{p}_{ij} = \ln p_{ij} + X_{i\cdot} \beta_{\cdot j} + \alpha_i + \epsilon_{ij}, \text{ for } \Delta_{ij} > 0, \quad (3)$$

so that the j th column of β contains the regression parameters for the bias factor of the j th taxon. We can either choose each α_i by minimizing $\sum_{j=1}^J (\ln \tilde{p}_{ij} - \ln p_{ij} - X_{i\cdot} \beta_{\cdot j} - \alpha_i)^2$ or else simply note that right-multiplying the row vector $\ln \tilde{p}_{i\cdot}$ as given in (3) by the compositional projection operator $P_i = \text{Diag}(\Delta_{i\cdot}) - \frac{1}{n_i} \Delta_{i\cdot}^T \Delta_{i\cdot}$ (where n_i is the number of taxa present in sample i) eliminates any term that is constant in i . By either approach, we obtain

$$Y_{i\cdot} = X_{i\cdot} \beta P_i + e_{i\cdot}, \text{ for } \Delta_{ij} > 0, \quad (4)$$

where $Y_{i\cdot} = (\ln \tilde{p}_{i\cdot} - \ln p_{i\cdot}) P_i$ and $e_{i\cdot} = \epsilon_{i\cdot} P_i$. We take $\Delta_{ij} \ln p_{ij} = 0$ if both Δ_{ij} and p_{ij} are equal to zero so that $Y_{ij} = 0$ if $\Delta_{ij} = 0$.

We propose to use least squares to estimate the parameters β in (4). We note however that there are more bias factor parameters (β) in (4) than we can estimate, corresponding to the fact that we can replace each β_j with $\beta_j + \beta_0$ in (1) with no change in any observable quantity. Equivalently, this can be seen as a consequence of the compositional constraint that the Y_{ij} values, summed over j , equal zero for each observation i . One way to account for this overparameterization would be to reparameterize in terms of a set of bias factor parameters β that are all identifiable. For example, in the simple model of Eq. (2), we could select one taxon (say, taxon J) to have $\beta_J = 0$, in which case the remaining β_j s would be interpreted as the difference between the log-bias factor for the j th taxon and the reference (J th) taxon. Another approach is to continue with the model with all of the β_j parameters but restrict inference to combinations of parameters that are identified. We choose this second, non-full-rank model because, in complex settings, it can be difficult to test complex hypotheses when parameters have been redefined. The cost in this choice is that we must exercise caution when performing inference to ensure that we only consider estimable combinations of parameters. A third approach in which constraints are added to the estimation procedure (i.e., $\sum_{j=1}^J \beta_j = 0$ for the simple model in (2)) is equivalent to our approach, as the least-squares estimators we use will automatically impose these constraints.

In order to obtain least-squares estimators, we vectorize the data from the i th sample using the vec trick to write

$$\text{vec}(Y_{i\cdot}) = P_i \otimes X_{i\cdot} \text{vec}(\beta) + \text{vec}(e_{i\cdot}),$$

where $\text{vec}(Q)$ is the column vector obtained by stacking the successive columns of Q , and \otimes denotes the Kronecker product. This gives a useful description of the data from the i th sample. If we further stack data from the 1st, 2nd, \dots , n th samples, we find

$$\begin{pmatrix} \text{vec}(Y_{1\cdot}) \\ \text{vec}(Y_{2\cdot}) \\ \vdots \\ \text{vec}(Y_{N\cdot}) \end{pmatrix} \equiv \text{vec}(Y^T) = \begin{pmatrix} P_1 \otimes X_{1\cdot} \\ P_2 \otimes X_{2\cdot} \\ \vdots \\ P_N \otimes X_{N\cdot} \end{pmatrix} \text{vec}(\beta) + \text{vec}(e^T) \equiv \mathbb{X} \text{vec}(\beta) + \text{vec}(e^T). \quad (5)$$

In this form, we see immediately that

$$\text{vec}(\hat{\beta}) = \mathbb{X}^- \text{vec}(Y^T), \quad (6)$$

where \mathbb{X}^- denotes the Moore–Penrose generalized inverse of matrix \mathbb{X} . For later use, we note that we can use the transpose property $(A \otimes B)^T = A^T \otimes B^T$ and the mixed-product property $(A \otimes B)(C \otimes D) = AC \otimes BD$ to write

$$\mathbb{X}^T \mathbb{X} = \sum_i \left\{ P_i \otimes X_{i\cdot}^T \right\} \{P_i \otimes X_{i\cdot}\} = \sum_i P_i \otimes \left(X_{i\cdot}^T X_{i\cdot} \right).$$

We further note that, for large sample sizes, it is easier to solve

$$\mathbb{X}^T \mathbb{X} \text{vec}(\hat{\beta}) = \mathbb{X}^T \text{vec}(Y^T)$$

to obtain $\text{vec}(\hat{\beta}) = (\mathbb{X}^T \mathbb{X})^- \mathbb{X}^T \text{vec}(Y^T)$, which is equivalent to Eq. (6).

2.3 Inference

The general form of a null hypothesis of interest is

$$\mathbb{C} \text{vec}(\beta) = 0, \quad (7)$$

where \mathbb{C} is a $D \times MJ$ matrix. We generally assume the contrasts that comprise the D rows of \mathbb{C} are linearly independent that D is the degrees of freedom of the hypothesis. For example, in an experiment in which all samples have the same three taxa and no additional covariates, we have $\beta = (\beta_{11}, \beta_{12}, \beta_{13})$, and hence, $\text{vec}(\beta) = \beta^T$, so we could test if the three bias factors were equal using the matrix

$$\mathbb{C} = \begin{pmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}.$$

In many contexts, hypotheses in Eq. (7) correspond to a matrix format that $C\beta Q = 0$, and in such cases, $\mathbb{C} \equiv (Q^T \otimes C)$. In this setup, Q is a fixed matrix that accounts for non-identifiability in β if the non-full-rank approach is used (as presented in this chapter). For example, choosing Q to be the $MJ \times (MJ - 1)$ -dimensional matrix with columns $I_{\cdot j} - I_{\cdot j'}$, $j \neq j'$ when writing βQ (where I is the identity matrix)

has the effect of subtracting $\beta_{\cdot j'}$ from every column of β and then removing the j' th column of the resulting matrix. This has the effect of choosing j' to be a reference taxon. In the example just considered, if we further choose C to be the 1×1 matrix with $C = 1$, then \mathbb{C} is the matrix given above, with $j' = 1$. As we generally use an overparameterized model, it is important to note that not every linear combination of $\text{vec}(\beta)$ is testable; we discuss this in Sect. 2.4.

We propose to use an F -statistic to evaluate statistical significance. To accomplish this, it is necessary to estimate β under the null hypothesis, i.e., subject to constraints in Eq. (7). This estimate, which we denote by $\hat{\beta}_0$, is easily obtained by solving a constrained least-squares problem; we use the R package `lse1` [17]. As the null model is nested within the full model, we have

$$F = \frac{RSS_0 - RSS}{RSS} = \frac{\sum_i \sum_j (r_{0,ij}^2 - r_{ij}^2)}{\sum_i \sum_j r_{ij}^2}, \quad (8)$$

where RSS_0 and RSS are the residual sums of squares under the null and full models, and where the full model residuals are given by

$$r_{i\cdot} = Y_{i\cdot} - X_{i\cdot}\hat{\beta}P_i, \quad (9)$$

and the null model residuals are given by $r_{0,i\cdot} = Y_{i\cdot} - X_{i\cdot}\hat{\beta}_0P_i$. Note that both r_{ij} and $r_{0,ij}$ are zero if Δ_{ij} is zero. Asymptotic inference is challenging because the number of parameters in β ($M \times J$) is often fairly large while the number of samples in many mock community studies is frequently small. Thus, we propose a permutation approach to assess significance (for which reason we have excluded the degrees of freedom usually found in an F -statistic).

Any Monte Carlo hypothesis test must account for the correlation between residuals of taxa in the same sample, which are always present if only because of the compositional constraint. Simply permuting the entire vector of residuals from each sample is not possible when some taxa are missing from some samples, and fails completely for the simplest case of Eq. (2), as each permutation replicate is identical to the original dataset. Thus, for each sample, we propose to first decorrelate the residuals under the null model, permute the decorrelated residuals, recorrelate them, and add them back to the predicted values calculated under the null. The decorrelation process also scales the residuals to have a common variance, up to the constraint imposed by compositionality. In this way, we re-generate data that share the same structure as the original data but are known to follow the null hypothesis.

Because of the compositional constraint, the residuals for each sample sum to zero. If Σ is the (unknown) variance–covariance matrix of a set of residuals in the absence of the compositional constraint, then the variance–covariance matrix for the constrained residuals for the i th of these samples is $\Sigma_i \equiv P_i \Sigma P_i$. For this reason, the “decorrelated” residuals will have variance–covariance matrix P_i rather than the usual identity matrix. We discuss this further in the Appendix, where we

also present a novel estimator $\widehat{\Sigma}$ for the variance–covariance matrix Σ for use in calculating Σ_i that accounts for both the compositional constraint and the complex missingness structure of these data. Since the correlation structure P_i corresponds to exchangeable residuals, we can still permute the “decorrelated” residuals within each sample, thereby accounting for the pattern of missingness in each sample, even though the variance of residuals across samples with different numbers of taxa will vary since the diagonal elements of P_i are given by $1 - 1/n_i$. After permutation, we recorrelate the permuted residuals to restore their variance–covariance to Σ_i . The explicit operations of this permutation are detailed in the following algorithm.

Algorithm: Permutation procedure for statistical inference

Data: $\tilde{p}, p, \mathbb{X}, \Delta, \mathbb{C}$.

Result: A p -value for statistical inference.

1. Fit the full model and estimate $\widehat{\beta}$. Obtain residuals $r_{i\cdot}$ for all samples.
2. Fit the constrained model and estimate $\widehat{\beta}_0$. Obtain residuals $r_{0,i\cdot}$ for all samples.
3. Calculate the F -statistic.
4. Calculate the variance–covariance matrix of null model residuals $\widehat{\Sigma}_0$ and hence obtain $\widehat{\Sigma}_{0,i} = P_i \widehat{\Sigma}_0 P_i$.
5. For each i , calculate the decorrelated residuals $r_{0,i\cdot}^{(de)} = r_{0,i\cdot} \widehat{\Sigma}_{0,i}^{-\frac{1}{2}}$.
6. For each i , randomly permute the $r_{i\cdot}^{(de)}$ values that correspond to $\Delta_{i,j} = 1$ to obtain the permuted decorrelated residuals $r_{0,i\cdot}^{(de,p)}$.
7. For each i , recorrelate the permuted residuals to obtain $r_{0,i\cdot}^{(p)} = r_{0,i\cdot}^{(de,p)} \widehat{\Sigma}_{0,i}^{\frac{1}{2}}$.
8. Generate a permutation null replicate dataset using $\text{vec}(Y_{i\cdot}) = X_{i\cdot} \widehat{\beta} + r_{0,i\cdot}^{(p)}$.
9. Calculate the test statistic $F^{(p)}$ using the permutation null replicate dataset.
10. Repeat the previous steps 6 to 9 B times, obtaining the F -statistic $F_b^{(p)}$ from the b th replicate.
11. Calculate $p = \frac{1}{B} \sum_{b=1}^B [I(F < F_b^{(p)}) + \frac{1}{2} I(F = F_b^{(p)})]$.

2.4 Testability of the Hypothesis

Models (1), (3), and (4) are overparameterized, so that arbitrary linear combinations of the elements of β are not estimable. Although we can always achieve a full-rank model by applying an appropriate set of constraints, in complex situations, it may not be easy to ascertain these constraints. Instead, we check the testability of any contrast we wish to test by confirming that it has no component that lies outside the space spanned by the rows of \mathbb{X} . By writing the singular value decomposition (SVD) of $\mathbb{X} = \mathbb{L}\mathbb{D}\mathbb{R}^T$, where we choose the form of SVD in which \mathbb{D} has only the

nonzero singular values, we can easily see that a linear combination of parameters $\mathbb{C}_i \cdot \text{vec}(\beta)$ is only estimable if $\mathbb{C}_i \cdot (\mathbb{I} - \mathbb{R}\mathbb{R}^T) = 0$. Thus, the general requirement for testability of a hypothesis of the form in Eq. (7) is

$$\|\mathbb{C}(\mathbb{I} - \mathbb{R}\mathbb{R}^T)\|_F^2 = 0, \quad (10)$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm of a matrix.

2.4.1 Example: Testable Hypotheses for Main Effects

The simplest model beyond (1) and (2) is a situation with a single binary covariate Z , such as an experiment run on two different plates. Here, we may be interested in testing whether the bias factors differ across plates. We can code the design matrix X in many ways, e.g., choosing $X_{i \cdot} = (I[Z_i = 1], I[Z_i = 2])$ or $X_{i \cdot} = (1, I[Z_i = 2])$. In both cases, β has two rows; in the first coding, the first row of β are the log-bias factors for plate 1 and the second row of β are the log-bias factors for plate 2, while for the second coding, the first row of β are the mean log-bias factors and the second row comprises the differences between log-bias factors across the two plates. In the first coding, we would test the hypothesis that $\beta_{1j} - \beta_{1j'} = \beta_{2j} - \beta_{2j'}$ for $j \neq j'$, while in the second coding, we test $\beta_{2j} = \beta_{2j'}$ for $j \neq j'$. These tests are equivalent since the β_{2j} values in the second coding are differences between bias factors across the two plates.

For Z having more than 2 levels, we would require β to have a row for each level of Z . For example, the samples in the Brooks' data were run on six plates, so that a test of any plate effect would require β to have six rows. The resulting design matrix is a $N \times 6$ matrix with each column indicating the plate membership. In principle, Z may also be continuous, in which case the corresponding rows of β would have the interpretation of the change in log-bias factor per unit change in the continuous covariate.

Panel A in Table 2 lists the testable hypotheses for main effects that we use in our simulations and in our analyses of the Brooks' dataset. Using the example of plate effects, H1 tests for the existence of any bias on any plate. H2–H4 focus on a single plate k and compare its log-bias factors to: the average log-bias factors in all other plates (H2), a constant value (H3), and the log-bias factors of another plate k' (H4), respectively. H5 switches gears and tests the differences in the bias factors of two taxa across all plates.

2.4.2 Example: Testable Hypotheses for Interaction Effects

In general, interactions between covariates Z can be treated as if they were “main effects” handled using the approach described in Sect. 2.4.1. In this section, we consider possible interactions *between taxa*. This is the central question in the Brooks' data, where we wish to test the hypothesis that bias factors do not depend

Table 2 Testable hypotheses used in simulations and in analyses of the Brooks data

<i>Hypotheses about main effect</i>			<i>Simulations</i>
<i>A</i>	<i>Research question</i>	<i>Null^a</i>	<i>Conditions that satisfy the null^b</i>
H1	Is there any bias across all plates and all taxa?	$\beta_{kj} - \beta_{kj'} = \beta_{k'j} - \beta_{k'j'}, \forall j \neq j', k \neq k'$	$b_1 = b_2 = 0$
H2(k)	Is the bias in plate k different from the average bias of all other plates, across all taxa?	$\beta_{kj} - \beta_{kj'} = \frac{1}{5} \sum_{k' \neq k} (\beta_{k'j} - \beta_{k'j'}), \forall j \neq j'$	$b_1 = b_2$
H3(k)	Is there any bias in plate k ?	$\beta_{kj} - \beta_{kj'} = 0, \forall j \neq j'$	$b_1 = b_2$
H4(k, k')	Is there any difference in the bias on plates k and k' , across all taxa?	$\beta_{kj} - \beta_{kj'} = \beta_{k'j} - \beta_{k'j'}, \forall j \neq j'$	$b_1 = b_2 = 0$
H5(j, j')	Is the relative bias between taxa j and j' the same across all plates?	$\beta_{kj} - \beta_{kj'} = \beta_{k'j} - \beta_{k'j'}, \forall k \neq k'$	$b_1 = b_2 = 0$
<i>Hypotheses about interaction effects</i>			<i>Simulations</i>
<i>B</i>	<i>Research question</i>	<i>Null^c</i>	<i>Conditions that satisfy the null</i>
H6(j, j')	Do taxa j and j' have the same interaction effect on the bias of the other taxa?	$\beta_{jk} - \beta_{j'k} = \beta_{jk'} - \beta_{j'k'} \text{ for } (k, k') \notin (j, j')$	$c_1 = c_2$
H7(j)	Does taxon j impact the bias of other taxa?	$\beta_{jk} - \beta_{jk'} = 0 \text{ for all } k, k' \text{ that } k \neq k' \neq j$	$c_1 = c_2 = 0$
H8	Is there any interaction effect among the taxa?	$\beta_{jk} - \beta_{jk'} = 0 \quad \forall j \neq k \neq k'$	$c_1 = c_2 = 0$

^a $k, k' \in (1, \dots, 6)$, the number of plates, $j, j' \in (1, \dots, 7)$ the number of taxa

^b b_1, b_2, c_1 , and c_2 are defined in Sect. 3

^c $k, k', j, j' \in (1, \dots, 7)$ the number of taxa

on the sample composition, i.e., that having taxon j in the sample has no effect on the (relative) bias of taxon k . To test this hypothesis, we take the vector of covariates to be $X_i = (1, \Delta_{i1}, \Delta_{i2}, \dots, \Delta_{iJ})$, where we recall $\Delta_{ij} = 1$ if sample i contains OTU j and $\Delta_{ij} = 0$ otherwise. The β matrix has the form

$$\beta = \begin{pmatrix} \beta_1 & \beta_2 & \cdots & \beta_J \\ 0 & \beta_{12} & \cdots & \beta_{1J} \\ \beta_{21} & 0 & \cdots & \beta_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{J1} & \beta_{J2} & \cdots & 0 \end{pmatrix}, \quad (11)$$

where each parameter β_j in the first row of Eq. (11) governs the intrinsic bias factor of taxon j , while each parameter β_{jk} gives the effect on the bias factor of taxon k due to the presence of taxon j in the sample. There is no reason to expect β to be symmetric.

Because some elements of β in Eq. (11) are set to zero by design, fitting this model would require extra machinery; for example, we could use constrained least

squares with J constraints to ensure $\beta_{(j+1)j} = 0$, $j = 1, \dots, J$. However, it is easy to see that Eq. (11) with X_i , as given above is equivalent to a model in which the vector of covariates is $X_i = (\Delta_{i1}, \Delta_{i2}, \dots, \Delta_{iJ})$ and

$$\beta = \begin{pmatrix} \beta_1 & \beta_{12} & \cdots & \beta_{1J} \\ \beta_{21} & \beta_2 & \cdots & \beta_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{J1} & \beta_{J2} & \cdots & \beta_J \end{pmatrix}, \quad (12)$$

where the parameters β_j now appear on the diagonal elements. This is possible because the intrinsic bias parameter β_j only applies to a sample containing OTU j , so the intercept in X_i , in the original formulation is not necessary. To add sample-level covariates such as plate effects to this model, we would add extra rows to the bottom of either (11) or (12) of the form described in Sect. 2.4.1.

The constraint that taxon j does not affect the bias factors of the other taxa present in the sample is that $\beta_{jk} = \beta_{jk'}$ for $j \neq k$, $j \neq k'$, $k \neq k'$. This corresponds to H7(j) in Table 2. For the Brooks' data with $J = 7$ taxa, this condition can be expressed using a constraint matrix \mathbb{C} having 5 rows; an easy way to write the rows is to pick some reference taxon $k \neq j$ and then let each row correspond to testing one constraint $\beta_{jk'} - \beta_{jk} = 0$ for $k' \neq k \neq j$. Across all taxa, there are $J \times (J - 2) = 7 \times 5 = 35$ contrasts corresponding to the interaction parameters. Testing them all corresponds to H8 in Table 2. There are in total $J^2 = 49$ parameters in (11) and (12). The null (column) space of \mathbb{X} is spanned by the 7 linear combinations of the $J = 7$ rows of (12) with coefficients equal to one and a linear combination of the diagonal elements of (12) with coefficients equal to one. The interaction effect corresponds to $J \times (J - 2) = 35$ constraints. The remaining $J - 1$ contrasts test $\beta_k + \sum_{j \neq k}^J \beta_{jk} = \beta_{k'} + \sum_{j \neq k'}^J \beta_{jk'}$, corresponding to testing that the total bias in OTU k equals the total bias in OTU k' in a sample with all OTUs present. It is not hard to show that if the interaction constraints are in force, these final contrasts correspond to testing the equality of the intrinsic bias parameters β_j . Finally, H6 in Table 2 tests whether two taxa have the same interaction effect, i.e., if they produce the same effect on the bias factors of the other taxa.

Another type of interaction (not considered further here) tests if the *relative abundance* of one taxon affects the bias factors of the other taxa. We would then code β as in Eq. (11) and replace Δ_i by p_i when constructing X_i for this analysis.

3 Simulations

We conducted simulations to assess the performance of our model under multiple realistic scenarios. Data were simulated to test both main effects as discussed in Sect. 2.4.1 and interactions as discussed in Sect. 2.4.2. All simulations followed the same true relative abundances and presence-absence pattern of taxa as in the

Brooks' dataset, to which we added bias using the log-linear model. Specifically, let p and p^* be, respectively, the $N \times J$ matrices of true and biased relative abundances (i.e., $p^* = E(\tilde{p})$). For each sample i , we simulated data using the following algorithm. First, we calculated a preliminary value for p_{ij}^* via

$$\ln(p_{ij}^*) = \ln(p_{ij}) + X_{ij}\beta, \text{ for } \Delta_{ij} \neq 0, \quad (13)$$

while choosing $p_{ij}^* = 0$ if $\Delta_{ij} = 0$. We then normalized the preliminary p_i^* values to sum to one. We then introduced random variation by generating read counts using a Dirichlet-multinomial distribution with expected proportion p_i^* , a dispersion parameter of 0.02, and a total read count of 5000. The dispersion parameter of 0.02 was selected to be similar to the estimated dispersion from a real microbiome dataset [3]. The observed data are the normalized read counts, denoted by \tilde{p} .

Note that p_i^* is not observed; further, p^* satisfies $p^* = E(\tilde{p})$ not $\ln p^* = E(\ln \tilde{p})$, so the data generation mechanism has the mean structure of model (1), not model (3). Thus, the simulation design provides an impartial assessment of our model even when the model assumptions do not hold. We considered multiple sample sizes ($N = 120, 240$, and 500). For $N = 240$, we simulated data that mimicked the data structure of the whole Brooks' dataset. For $N < 240$ or $N > 240$, we first randomly downsampled or upsampled values of Δ_i and p_i with replacement from the Brooks' data to a sample size N and then followed the same protocol for simulation. 5000 simulations were used to evaluate type I error, and 2000 simulations were used to evaluate the statistical power.

3.1 Main Effect Simulation

We conducted simulation studies to confirm the validity of our inference on main effects. In the Brooks' data, samples were processed through six plates. In this simulation, we coded the ($N \times 6$ -dimensional) design matrix X using six indicators for plate membership as described in Sect. 2.4.1. Then, β is a 6×7 matrix of coefficients, in which the k th row contains the log-bias factors for the k th plate. We used the following β matrix for simulation:

$$\beta = \begin{pmatrix} b_1 & b_1 & b_1 & b_1 & b_1 & b_2 & b_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_2 & b_2 & b_2 & 0 & 0 & 0 \\ 0 & b_2 & b_2 & b_2 & 0 & 0 & 0 \\ 0 & b_2 & b_2 & b_2 & 0 & 0 & 0 \end{pmatrix}. \quad (14)$$

There are only two free parameters in this matrix. In this simulation, the taxa relative abundances in plates 2 and 3 are unbiased. Depending on the values of b_1

and b_2 , some or all taxa on plates 1, 4, 5, and 6 may be biased compared to the true relative abundances. When $b_1 = b_2$, there is no bias in plate 1 even when b_1 and b_2 are not zero, because adding a constant to a row of β does not change $E(\tilde{p}_{ij})$. Plates 4, 5, and 6 have the same true bias structure. For this simulation setup, the ratio of bias factors for taxa 6–7 to taxa 1–5 is $\exp(b_2 - b_1)$ (e.g., when $b_2 = 0.3$ and $b_1 = 0.1$ as in one of our simulations, $\exp(b_2 - b_1) = 1.22$), and the ratio of bias factors for any pair of taxa among taxa 1–5 (or between taxa 6 and 7) is one in plate 1. The ratios of bias factors can be computed in a similar way for other plates and between other pairs of taxa.

For the main effect simulation, we evaluated four different hypotheses, viz. H1, H2(1), H3(1), H4(1,6), H5(1,2) in Table 2. The value in parentheses gives the specific plates (k or k') or taxa (j or j') that are tested. These hypotheses represent a diverse range of research questions that are of scientific importance, including tests of different experimental procedures and tests against different taxa. The last column of Table 2 gives the sufficient conditions under which the null hypotheses are satisfied in our simulation setup.

3.2 Interaction Effect Simulation Based on the Brooks Data

We conducted additional simulations to confirm the validity of our inference on interaction effects in the presence of main effects. These simulations follow the same procedure as in the main effects simulations, but with a different design matrix. We used X as a $N \times 12$ matrix, in which the first seven columns indicated the presence-absence of each taxon in each sample, and the last five columns of X indicated if the sample was processed on plate k , $k = 2, \dots, 6$. With this choice of X , β is a 12×7 matrix for which the first seven rows represent the interaction effect (and intrinsic bias of plate 1) as coded in Eq. (12), and the last five rows represent the main plate effects. In this simulation, the first seven rows of β take the following values:

$$\begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{17} \\ \beta_{21} & \beta_2 & \cdots & \beta_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{71} & \beta_{72} & \cdots & \beta_7 \end{pmatrix} = \begin{pmatrix} c_1 & c_1 & c_1 & c_1 & c_1 & c_2 & c_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_2 & c_2 & c_2 & c_2 & 0 \\ 0 & 0 & c_2 & c_2 & c_2 & c_2 & 0 \end{pmatrix}.$$

The last five rows of β are the same as the first five rows of Eq. (14) with $b_1 = 0.1$ and $b_2 = 0.3$. Note that the interpretations of the main effects in this simulation are slightly different from those in Sect. 3.1, as the last 5 rows of β represent differences between plate k and plate 1, for $k = 2, \dots, 6$.

In this simulation, taxa 2–5 do not interact with any other taxa. Depending on the values of c_1 and c_2 , taxon 1, 6, and 7 may interact with some or all of the other taxa. As in the main effect simulation, if $c_1 = c_2$, taxon 1 does not interact with other taxa even if c_1 and c_2 are not zero. If $c_1 \neq c_2 \neq 0$, then the presence of taxon 1 affects the bias factors of taxa 2–7. When $c_2 \neq 0$, the presence of taxon 6 affects the bias factors of taxa 3–5 and the presence of taxon 7 affects the bias of taxa 3–6. Note that even though the 6th and 7th rows of β are the same, their interpretations are slightly different because $\beta_{66} = c_2$ specifies the intrinsic bias of taxon 6 rather than an interaction, as discussed in Sect. 2.4.2. We evaluated three hypotheses in our simulation of interaction effect (H6(1,2), H7(1), H8 in Table 2).

4 Results

4.1 Simulation Results

Table 3 summarizes the type I errors for all our simulations. Our proposed permutation approach controls the type I errors for all simulations. When the sample size is small, some tests also show slightly conservative type I error, particularly H7 and H8 for sample sizes $N = 120$ or 240 . This may occur for a variety of reasons. First, the number of identifiable parameters in these simulations is large: 35 in the main effects simulations and 71 in the interaction simulations. Second, our permutation approach requires de-correlating the residual errors in each sample. This step requires a good estimator of the residual covariance matrix, which in turn requires a reasonable sample size. Finally, we also note that our data was

Table 3 Type I error simulation results

A: Main effect models								B: Interaction models				
N	b_1	b_2	H1	H2(1)	H3(1)	H4(1,6)	H5(1,2)	c_1	c_2	H6(1,2)	H7(1)	H8
120	0	0	0.048	0.045	0.045	0.051	0.042	0.0	0.0	0.042	0.035	0.034
120	0.1	0.1	—	—	0.051	—	—	0.1	0.1	0.046	—	—
120	0.3	0.3	—	—	0.052	—	—	0.3	0.3	0.048	—	—
240	0.0	0.0	0.048	0.040	0.045	0.041	0.044	0.0	0.0	0.057	0.039	0.031
240	0.1	0.1	—	—	0.048	—	—	0.1	0.1	0.056	—	—
240	0.3	0.3	—	—	0.038	—	—	0.3	0.3	0.058	—	—
500	0.0	0.0	0.046	0.042	0.037	0.048	0.044	0.0	0.0	0.052	0.045	0.044
500	0.1	0.1	—	—	0.043	—	—	0.1	0.1	0.053	—	—
500	0.3	0.3	—	—	0.042	—	—	0.3	0.3	0.053	—	—
1000	0.0	0.0	0.042	0.046	0.046	0.050	0.060	0.0	0.0	0.048	0.047	0.046
1000	0.1	0.1	—	—	0.040	—	—	0.1	0.1	0.049	—	—
1000	0.3	0.3	—	—	0.048	—	—	0.3	0.3	0.054	—	—

“—” indicates that the simulation setting does not belong to the null hypothesis

simulated based on a Dirichlet-multinomial distribution, rather than the log-linear model we assume in the analysis. It is not clear how important this last issue is; if it were a major contributor, we would expect the Haldane-like correction described after Eq. (2) to improve the performance of these tests. In limited simulations (results not shown), we found no improvement in test size when this correction was implemented.

Figure 1 shows the power achieved by our method in our simulations. For all hypotheses and methods, power increases when the sample size or effect size increases. The power values are not directly comparable across hypotheses because of the differences in the underlying (composite) null models. However, it is worth noticing some trends. For all the main effect hypotheses, H3 appears to be the least powerful because it only compares one plate with the truth, while the other hypotheses compare multiple plates and use more data. In the interaction tests, H8 is the most powerful because H8 tests “any” interaction, while the other hypotheses

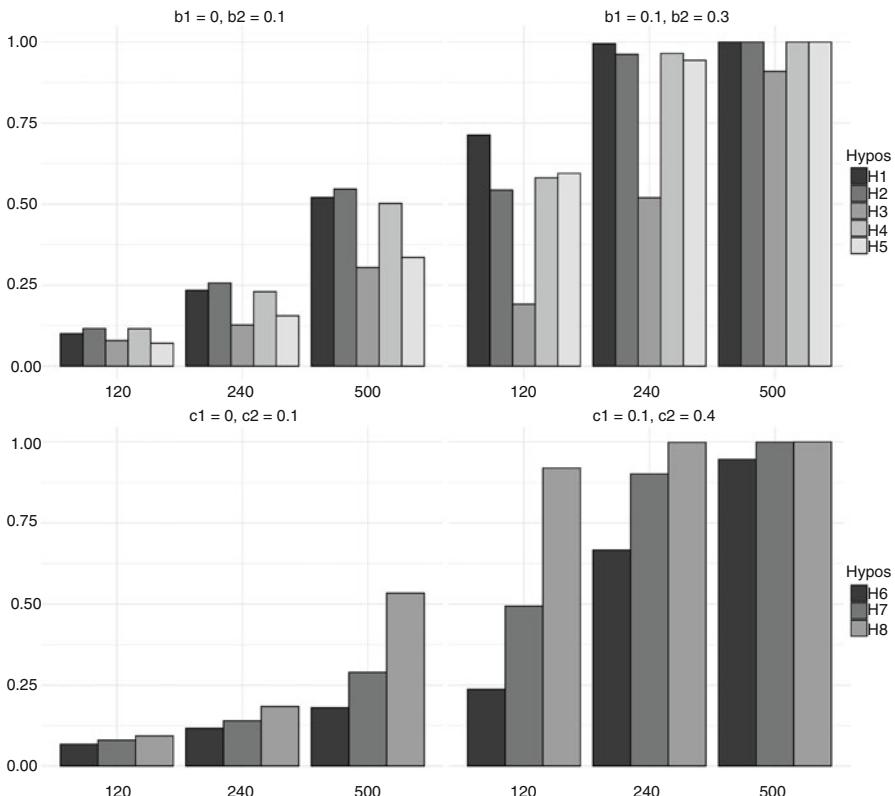


Fig. 1 Power result for simulation. Upper panel: main effect models. Lower panel: interaction effect models. x -axis: different sample sizes; y -axis: estimated power; color of the bar: different hypotheses

test only a subset of the possible interaction effects. It is possible to simulate data in which the selected signal would make one or another hypothesis more powerful. But in the current simulation in which the assumed interactions are distributed across taxa and plates, the superior power of the “omnibus” hypothesis H8 is as expected.

4.2 Do Interactions Between Taxa Affect Bias in the Brooks’ Data?

We applied our proposed methods to analyze the Brooks’ data. We adopted a backward elimination strategy, starting with a full model with both plate effects and taxon–taxon interactions. We then tested whether the interactions can be removed from the model. If the interaction effects are non-significant or much smaller compared to the main plate effects, we remove them from the model and proceed to assess the plate effects in more detail.

We first tested for evidence of taxon–taxon interactions on bias factors in the full data. We investigated whether each taxon individually impacts other taxa (hypotheses H7(1)–H7(7)) and whether there is any taxon–taxon interaction effect (H8). The design matrix X and coefficient matrix β were coded as described in Sect. 3.2. Table 4 shows the results for these analyses. In the full dataset, we found no evidence of interaction, with an overall interaction p -value of 0.793 (Panel A, Table 4).

Because of the substantial difference in sample types of samples, we also conducted separate analyses using data from plates 1 & 2 (cells), plates 3 & 4 (DNA), and plates 5 & 6 (PCR products). We tested the same hypotheses in each analysis as in the pooled analysis and found that there is a significant overall interaction effect in plates 1 & 2; three taxa also showed a significant interaction at the $\alpha = 0.05$ level (*L. crispatus*, *A. vaginae* and *GBS*). For the DNA samples, one taxon was significant at the $\alpha = 0.05$ level (*L. iners*), while one was marginal (*L. crispatus*), but the overall test H8 was not significant for either the DNA samples or the PCR products (Panel B, Table 4).

If we adjust to the $3 \times 7 = 21$ taxon-specific interaction tests to account for multiple testing, none of the interactions remain significant (Bonferroni p -value cutoff = 0.0016). However, the cell samples (plates 1 and 2) are inherently different from the other types of samples, as bacterial cells compete in the cell lysis and DNA extraction process, and therefore are more subject to taxon–taxon interaction. For this reason, we may wish to consider only applying the multiple comparison correction to the tests of interaction in plates 1 and 2; in this case, one taxon–taxon interaction (*GBS*) remains significant (Bonferroni p -value cutoff = 0.006). Given this discrepancy, it would be helpful to assess the relative importance of taxon–taxon interaction compared to the main effects of bias factors in plates 1 and 2.

In Panel C of Table 4, we show the estimated parameters in β for our analysis of plates 1 and 2. The first line gives the average main effect (intrinsic bias) for

Table 4 Interaction analysis results for the Brooks' data

A: Interaction test results using all data (p-values)						
	<i>L. crispatus</i>	<i>L. iners</i>	<i>G. vaginalis</i>	<i>A. vaginalae</i>	<i>P. bivia</i>	<i>S. amnii</i>
All plates	0.792	0.745	0.147	0.499	0.691	0.887
B: Interaction tests results in each stratum (p-values)						
Plates 1 & 2 (cell)	0.023	0.710	0.612	0.016	0.645	0.263
Plates 3 & 4 (DNA)	0.052	0.014	0.639	0.548	0.767	0.933
Plates 5 & 6 (PCR)	0.442	0.537	0.482	0.947	0.862	0.656
C: Estimated beta coefficients and their variances for stratified analysis using plates 1 & 2						
	<i>L. crispatus</i>	<i>L. iners</i>	<i>G. vaginalis</i>	<i>A. vaginalae</i>	<i>P. bivia</i>	<i>S. amnii</i>
Main (average)	0.702	1.396	-1.718	-1.169	0.530	1.384
Main (differences)	0.054	0.002	0.042	-0.087	0.168	-0.082
Taxon-taxon interaction effects						
<i>L. crispatus</i>	-	-0.122	-0.182	-0.004	-0.369	0.032
<i>L. iners</i>	-0.265	-	-0.143	-0.228	-0.205	-0.248
<i>G. vaginalis</i>	0.231	0.295	-	0.399	0.265	0.262
<i>A. vaginalae</i>	0.187	0.112	0.371	-	0.150	-0.033
<i>P. bivia</i>	-0.053	-0.074	-0.030	-0.194	-	-0.031
<i>S. amnii</i>	-0.233	-0.238	-0.107	-0.202	-0.252	-
<i>GBS</i>	0.208	0.044	0.234	0.448	0.094	0.099
						-
						0.052
						Variance
						GBS
						Overall

these two plates; the second line gives the difference in values of the main effect (differences in intrinsic bias) across the two plates. The remaining lines give the interaction terms, with rows corresponding to the taxon, the presence of which causes the bias, and columns corresponding to the taxon for which the bias is being altered. Recall that the bias factors remain the same if we add a constant to each row or to the diagonal elements of the matrix β . Therefore, it is the *variability* in the values of β , rather than their magnitude, that determines the effect size of the bias they represent. For this reason, in the final column of Panel C, we compare the variance of the values of β corresponding to the main effect (1.704), the variance of the values of β corresponding to the difference in main effects across the two plates (0.009), and the variance of the values of β corresponding to interaction effects (0.052). Note that the variance of the main (average) effects is nearly 33 times the variance of the interaction bias effects, indicating that even when the interaction effect is significant, its magnitude is much smaller than the main effect. Finally, note that this measure of effect size for the magnitude of the difference in main effects across the two plates is negligible; this is discussed further in Sect. 4.3.

It is also important to note that the results here are limited to the specific protocols used by Brooks et al. and may not hold for different protocols. In particular, since the interaction effect is limited to the cell samples, the interaction effect we observe is presumably a result of the extraction protocol used to lyse the cells to extract their DNA.

4.3 Plate and Sample Type Effects in the Brooks' Data

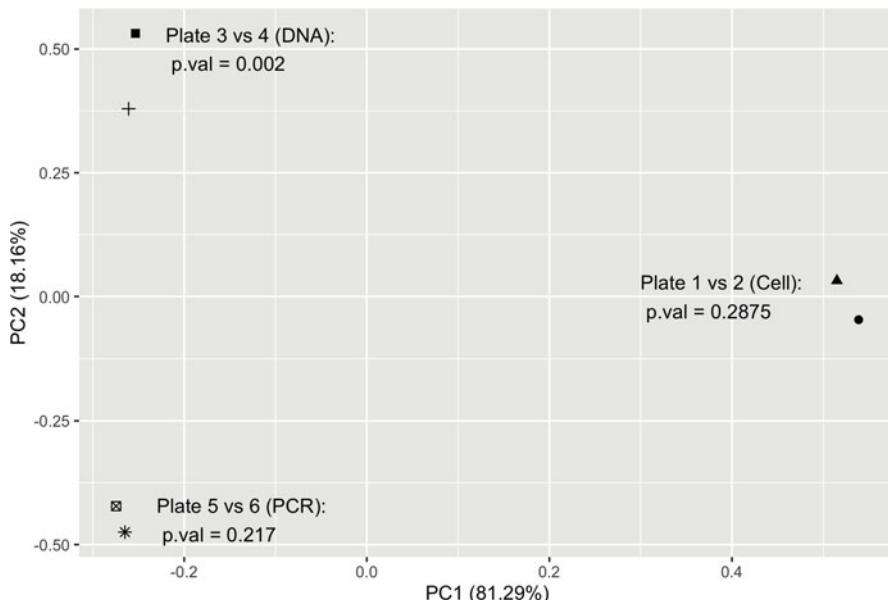
Given that the interaction effect is much smaller than the main effect, we conducted further analyses investigating the main effects of sample type and plate, removing interactions from the model. We coded the design matrix X as described in Sect. 3.1 with an indicator for each plate. For hypothesis testing, we tested hypotheses H1–H4 in Table 2 for all plates.

Table 5 gives the estimated β matrix, with each row showing the log-bias factors (chosen to sum to zero) for each plate. The β estimates for plates corresponding to the same sample type (1 & 2, 3 & 4, 5 & 6) are very similar, also indicating the plate effects within sample types are small. Using hypotheses H1–H3, we found very strong evidence of bias with all p -values $< 10^{-4}$. For H4, we conducted pairwise comparison for all plates: all comparisons are significantly different with p -values $< 10^{-4}$ except for the following pairs (plate 1 vs. 2, $p = 0.287$; plate 3 vs. 4, $p = 0.002$, and plate 5 vs. 6, $p = 0.217$). Figure 2 shows a two-dimensional ordination of the six plates based on their bias factors. The relative magnitudes of the effect of sample type and plate (within sample type) are easily visualized in this figure.

Table 5 Main effect model β estimates for the Brooks data

	<i>L. crispatus</i>	<i>L. iners</i>	<i>G. vaginalis</i>	<i>A. vaginae</i>	<i>P. bivia</i>	<i>S. amnii</i>	<i>GBS</i>
Plate 1	0.843	1.557	-1.832	-1.295	0.654	1.482	-1.409
Plate 2	0.807	1.535	-1.829	-1.212	0.489	1.565	-1.355
Plate 3	-0.738	0.858	-0.948	0.091	-0.964	1.023	0.679
Plate 4	-0.520	0.828	-0.877	0.021	-0.903	0.743	0.709
Plate 5	-0.089	0.242	-0.165	0.062	-0.167	0.269	-0.152
Plate 6	-0.102	0.121	-0.260	-0.014	0.028	0.258	-0.030

PCA of beta matrix

**Fig. 2** Main effect model β estimates for the Brooks data

5 Discussion

Bias is ubiquitous in all microbiome sequencing studies. Depending on the experimental and analysis protocols, the measured relative abundance can differ by an order of magnitude or more, even when the true relative abundances are the same [4, 9]. For example, in the Brooks' data, the ratio of bias factors for *L. crispatus* to *GBS* is approximately 9.1 in cell samples (calculated by exponentiating the average log-bias ratio between the two taxa in the top panel of Fig. 2) but is approximately 1.0 in PCR product samples. These biases pose a threat to the reliability and reproducibility of microbiome studies.

The extent to which biases in relative abundances, if unaccounted for, affect the conclusion of an analysis may depend on the question being asked. Statistical

inferences that directly rely on an “effect estimate,” such as differential abundance analysis, regression models, and mediation analysis may be completely invalidated by bias if it is not accounted for. MWC supports this claim by demonstrating that even when all samples have the same bias factors (e.g., when they were processed by the same lab using the same protocol), the biases do not cancel out (Figure 2 in [9]). Until bias in data from microbiome studies can be properly accounted for, any conclusions on individual taxa relative abundances that cannot be validated by other means (such as animal models) will be suspect. For classification or cluster analysis, the impact of bias may be milder if similar samples are affected by bias in a similar way; the effect of bias for these types of analyses should be investigated. When all samples have exactly the same bias factors, the compositional distance between samples, defined by Aitchison et al. [1], remains the same; however, use of the compositional distance is difficult when some taxa have zero counts in some samples, as is typical in microbiome studies. Other commonly used distance measures, such as the Bray–Curtis or weighted UniFrac distances, are not unaffected by bias factors and so may be a threat to the validity of distance-based analyses even when bias factors do not vary across samples.

Modeling bias is challenging because the sources of bias are widespread through the entire experimental/analysis pipeline and are both protocol- and taxon-dependent. It is almost impossible to individually account for all verified sources of bias. The MWC model [9] offers a great simplification in studying the bias generation process and provides a language for how bias can be described. By demonstrating that sources of bias act in a multiplicative fashion, MWC model allows us to focus on bias factors for individual taxa, and how these bias factors depend on experimental protocols. Here, we have extended the MWC model by writing it as a log-linear model for taxon relative abundances and including covariates to explicitly model the bias introduced by sample-specific characteristics. These methods allow us to answer complex questions about mock community data, in which the true relative abundance is known. In our analysis of the Brooks’ dataset, we found some evidence for taxon–taxon interactions in plates 1 and 2 (cell samples), showing the capacity of our model to extend the MWC model. However, we also showed that the effect size for interaction was much smaller than the main effect, explaining why their original model performed very well in predicting the true taxon relative abundances. Further, as these results only apply to the cell samples, they presumably depend on the particular DNA extraction method used and may not generalize to other extraction methods. This suggests another use for our approach, comparing extraction protocols to minimize or eliminate taxon–taxon interaction. Finally, the results presented here require knowledge of the true relative abundances of each taxon; for a limited extension to a situation where true taxa relative abundances are not known, see Tyx et al. [15].

Our permutation framework is very flexible. It can be used when all taxa are present in all samples, as well as when samples have different missing data structures. It performed well for all the hypotheses that we evaluated. Our approach requires a reasonable estimator of the residual variance–covariance matrix ($\widehat{\Sigma}_i$). The number of elements in this matrix increases as the square of the number of

taxa, which can make estimation of $\widehat{\Sigma}_i$ difficult when the sample size is small. To investigate this, we also simulated larger microbiome mock communities with 14 and 21 OTUs (data not shown) and demonstrated that our permutation approach performed reasonably well with valid type I error. For situations with a larger number of taxa, it may be worth considering a shrinkage estimator of $\widehat{\Sigma}_i$, described briefly in the Appendix.

In the Brooks' data, the library sizes differ significantly across plates (Table 1); however, the differences are small, and all samples have large library sizes. As the relative abundances of all taxa present in any sample in these data are large, the observed library sizes permit a reasonably precise measurement of the relative abundances, so the differences in library sizes should have little impact on the performance of our log-linear model. This is the case in simple mock community experiments (such as in the Brooks' data) when the total number of taxa is small. In situations when library sizes differ dramatically, or in mock communities with many taxa, some of which are rare, additional heteroscedasticity may be introduced. Even in such a context, the methods proposed here for estimation and inference are still valid. The library sizes impact the precision of the observed relative abundances, whose effect manifests through the variance of the residuals. Our least-squares estimators are agnostic to the residual covariances. Further, differences in library size are attenuated by working on the log relative abundance scale. Our inference remains valid because of our permutation framework: we permute the residuals corresponding different taxa within samples, which are not impacted by the library sizes. For the same reason, our inference is valid with unbalanced number of taxa across samples. Further methodology research that explicitly takes into account the library sizes, possibly via some inverse-precision weighting framework, can potentially improve the power of our model.

Mock community samples will typically comprise no more than 20 taxa for technical reasons (Scott Jackson, National Institute of Standards and Technology, personal communication). Thus, it is of interest to ask whether bias factors can be estimated for, say, 50 taxa using three model communities (say A, B, and C) each having 20 taxa, in which A and C have no taxa in common but B has 5 taxa in common with both A and C. This question is important if we wish to develop a large library of (relative) bias factors. Using the methods we have developed, we find that it is not necessary for each possible pairing of taxa to be observed in some sample, as the relative bias of taxa 1 and 2 can be combined with the relative bias of taxa 2 and 3 to infer the relative bias of taxa 1 and 3. A related question is what types of samples are required to learn about interactions between taxa. As we have discussed, it is necessary to have samples with a different number of taxa; empirically, we find that even if we only see samples having either m_1 taxa or m_2 taxa, we can still identify all the interaction parameters as long as we see "full sets" (i.e., all possible combinations of m_1 taxa and all possible combinations of m_2 taxa). In fact, we were surprised to find that even choosing $m_1 = 2$ and $m_2 = 3$ allowed identification of the interaction effects. For these "experiments," we only considered "full sets"; it

may be possible to develop designs that identify all interactions but use a smaller set of combinations of taxa.

A persistent problem in microbiome studies is the lack of an accepted way to normalize or standardize microbiome data. There appear to be three possible approaches to handle bias. The first is to improve the laboratory and bioinformatic methods so that the relative abundance data they produce is unbiased. The second is to develop statistical methods that give unbiased results even in the presence of experimental bias. For example, all relative abundance measures could be reported as a fraction of the relative abundance found in a “standard” sample. A small modification of our version of the MWC model can provide a third approach to this problem. If a set of taxon-level covariates could be found that would describe the variability of bias factors throughout the microbiome, then we could fit this model and use the results to divide each observed taxon read count by an estimate of its bias factor. Examples of such covariates might include Gram status, cell wall composition, measures of primer mismatch, and GC content. To fit a model such as this, we would need mock community data with known relative abundances, as well as a vector of L taxon-specific covariates. If we let $Z^{(j)}$ be the $M \times L$ -dimensional matrix corresponding to the covariates for the bias factors of taxon j , we could then relate the log-bias factors β_j to taxon-level covariates using the model

$$\beta_j = Z^{(j)}\gamma, \quad (15)$$

where Z is a design matrix for the taxon-level model in which the j th row contains the covariates that describe the j th taxon. Recalling the definition of the vec operator, Eq. (15) implies $\text{vec}(\beta) = \mathbb{Z}\gamma$, where

$$\mathbb{Z} = \begin{pmatrix} Z^{(1)} \\ \vdots \\ Z^{(J)} \end{pmatrix},$$

so we can rewrite Eq. (5) in Sect. 2.2 as

$$\text{vec}(Y^T) = \mathbb{X}\mathbb{Z}\gamma + \text{vec}(e^T),$$

where, in a slight abuse of notation, we have used the same notation for the error term. The least-squares estimators of γ are then easily found to be $\hat{\gamma} = (\mathbb{X}\mathbb{Z})^{-}\text{vec}(Y^T)$. One could imagine a procedure in which this model was fit to an appropriate mock community dataset run on each plate to give a plate-specific bias factor. Note that several assumptions would be required for this method to be workable. Most importantly, either the covariates in Z would have to be easily available for every microbe in a sample or else a reasonably simple way to impute these covariates when they are not available would be required. One possible imputation model may be to assume that covariates segregate as traits in

a phylogenetic tree. We believe progress implementing this program could be an important step in making microbiome studies reproducible and reliable.

Acknowledgments NZ is supported in part by the National Institutes of Health, Environmental Influences of Child Health Outcomes (ECHO) Data Analysis Center (U24OD023382). GS is supported in part by the National Institutes of Health, National Institute of Environmental Health Sciences (R24ES029490) and the Office of the Director (UG3OD023318/UH3OD023318).

Appendix

From Eq. (9) and the definition of $Y_{i\cdot}$ after Eq. (4), we see that the form of the variance–covariance matrix of the residuals for sample i is $\Sigma_i = P_i \Sigma P_i$. If the compositional mean is known, denoted by μ , a simple estimator for Σ_i is to first estimate Σ by solving the estimating equation

$$\widehat{V} := \sum_{i=1}^N (r_{i\cdot}^T - P_i \mu)(r_{i\cdot} - \mu^T P_i) = \sum_i P_i \Sigma P_i. \quad (16)$$

For each i in the sum, we use the vec trick and then solve the resulting equation for Σ to obtain

$$\text{vec}(\widehat{\Sigma}) = \left(\sum_i P_i \otimes P_i \right)^{-1} \text{vec}(\widehat{V}). \quad (17)$$

If there is a reason to believe that there is substantial variation in the precision of the data across samples (which may occur if the variation in library sizes across samples is large enough), we may wish to weight the terms in the sums of Eq. (16) by weights ω_i that are proportional to the precision of the data from the i th sample. We have not considered this as the large library sizes in the Brooks' data would seem to make this unnecessary.

In general, the centering vector μ is unknown and needs to be estimated. $\widehat{\mu}$ can be obtained by solving the estimating equation

$$\sum_i r_{i\cdot}^T = \sum_i P_i \mu, \quad (18)$$

which has solution

$$\widehat{\mu} = \left(\sum_i P_i \right)^{-1} \left(\sum_i r_{i\cdot}^T \right).$$

As with the estimator of Σ , we may wish to weight the sums in Eq. (18) if there is a substantial difference in precision across samples.

Estimation of the compositional mean $\widehat{\mu}$ and variance $\widehat{\Sigma}$ was considered by van den Boogart and Tolosana-Delgado [16]. Here, we use the same estimator $\widehat{\mu}$ as [16], but use the novel estimator for $\widehat{\Sigma}$ shown here because the estimator derived in [16] is more complex and slower to compute. We typically find $\widehat{\mu} = 0$, except in cases where the null model does not allow for a separate intercept for each feature (taxon).

If the number of taxa is large, a shrinkage estimator of \widehat{V} can be used. This will in turn imply a shrinkage estimator of $\widehat{\Sigma}$ via Eq. (17). One possible shrinkage approach is the empirical Bayes shrinkage proposed by Ledoit and Wolf [8], which was implemented in R package “CovTools” [18]. In this approach, Σ is estimated using $\delta\widehat{\Sigma} + (1 - \delta)T$, where $\widehat{\Sigma}$ is the estimated variance–covariance matrix (e.g., as estimated as in Eq. (17)) and T is a pre-defined target matrix. In situations when the residuals are full rank, the target matrix is usually taken as the identity matrix or a diagonal matrix with positive diagonal elements. In the current context, a reasonable target matrix can be $\hat{\sigma}^2 \sum_i P_i/n$, in which $\frac{1}{n} \sum_i P_i$ is the average of the compositional projection operators, and $\hat{\sigma}^2$ is the usual variance estimated from $|r_i^T - P_i \widehat{\mu}|$.

Finally, we note that since the decorrelated residuals are given by $\widehat{\Sigma}_i^{-\frac{1}{2}} r_i^T$, their variance–covariance matrix is $(P_i \widehat{\Sigma} P_i)^{-\frac{1}{2}} (P_i \widehat{\Sigma} P_i) (P_i \widehat{\Sigma} P_i)^{-\frac{1}{2}}$ (under the assumption that Σ is well estimated). Using the SVD to express $P_i \widehat{\Sigma} P_i$, it is easy to see this variance–covariance matrix is just the projection operator into the range (column or row space) of $P_i \widehat{\Sigma} P_i$. By assumption, we take the range of $\widehat{\Sigma}$ to contain the range of P_i ; thus, this projection operator is in fact P_i itself.

References

1. Aitchison, J., Barceló-Vidal, C., Martín-Fernández, J., Pawlowsky-Glahn, V.: Logratio analysis and compositional distance. *Math. Geol.* **32**, 271–275 (2000)
2. Brooks, J.P., Edwards, D.J., Harwiche, M.D., Rivera, M.C., Fettweis, J.M., Serrano, M.G., Reris, R.A., Sheth, N.U., Huang, B., Girerd, P., Strauss, J.F., Jefferson, K.K., Buck, G.A.: The truth about metagenomics: quantifying and counteracting bias in 16S rRNA studies. *BMC Microbiol.* **15**, 66 (2015)
3. Charlson, E.S., Chen, J., Custers-Allen, R., Bittinger, K., Li, H., Sinha, R., Hwang, J., Bushman, F.D., Collman, R.G.: Disordered microbial communities in the upper respiratory tract of cigarette smokers. *PLoS One* **5**(12), e15216 (2010)
4. Costea, P.I., Zeller, G., Sunagawa, S., Pelletier, E., Alberti, A., Levenez, F., Tramontano, M., Driessens, M., Hercog, R., Jung, F.E., Kultima, J.R., Hayward, M.R., Coelho, L.P., Allen-Vercoe, E., Bertrand, L., Blaut, M., Brown, J.R.M., Carton, T., Cools-Portier, S., Daigneault, M., Derrien, M., Druesne, A., de Vos, W.M., Finlay, B.B., Flint, H.J., Guarner, F., Hattori, M., Heilig, H., Luna, R.A., van Hylckama Vlieg, J., Junick, J., Klymiuk, I., Langella, P., Le Chatelier, E., Mai, V., Manichanh, C., Martin, J.C., Mery, C., Morita, H., O'Toole, P.W., Orvain, C., Patil, K.R., Penders, J., Persson, S., Pons, N., Popova, M., Salonen, A., Saulnier, D., Scott, K.P., Singh, B., Slezak, K., Veiga, P., Versalovic, J., Zhao, L., Zoetendal, E.G., Ehrlich, S.D., Dore, J., Bork, P.: Towards standards for human fecal sample processing in metagenomic studies. *Nat. Biotechnol.* **35**(11), 1069–1076 (2017)

5. D'Amore, R., Ijaz, U.Z., Schirmer, M., Kenny, J.G., Gregory, R., Darby, A.C., Shakya, M., Podar, M., Quince, C., Hall, N.: A comprehensive benchmarking study of protocols and sequencing platforms for 16S rRNA community profiling. *BMC Genom.* **17**, 55 (2016)
6. Hugerth, L.W., Andersson, A.F.: Analysing microbial community composition through amplicon sequencing: from sampling to hypothesis testing. *Front. Microbiol.* **8**, 1561 (2017)
7. Kembel, S.W., Wu, M., Eisen, J.A., Green, J.L.: Incorporating 16S gene copy number information improves estimates of microbial diversity and abundance. *PLoS Comput. Biol.* **8**(10), e1002743 (2012)
8. Ledoit, O., Wolf, M.: Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *J. Empir. Finan.* **10**(5), 603–621 (2003) ISSN: 0927-5398
9. McLaren, M.R., Willis, A.D., Callahan, B.J.: Consistent and correctable bias in metagenomic sequencing experiments. *eLife* **8**, e46923 (2019). ISSN: 2050-084X
10. Morgan, J.L., Darling, A.E., Eisen, J.A.: Metagenomic sequencing of an in vitro-simulated microbial community. *PLoS One* **5**(4), e10209 (2010)
11. Pollock, J., Glendinning, L., Wisedchanwet, T., Watson, M.: The madness of microbiome: attempting to find consensus “Best Practice” for 16S microbiome studies. *Appl. Environ. Microbiol.* **84**(7), e02627-17 (2018)
12. Ross, M.G., Russ, C., Costello, M., Hollinger, A., Lennon, N.J., Hegarty, R., Nusbaum, C., Jaffe, D.B.: Characterizing and measuring bias in sequence data. *Genome Biol.* **14**(5), R51 (2013)
13. Siegwald, L., Caboche, S., Even, G., Viscogliosi, E., Audebert, C., Chabé, M.: The impact of bioinformatics pipelines on microbiota studies: does the analytical “Microscope” affect the biological interpretation? *Microorganisms* **7**(10), 393 (2019)
14. Sinha, R., Abu-Ali, G., Vogtmann, E., Fodor, A.A., Ren, B., Amir, A., Schwager, E., Crabtree, J., Ma, S., Abnet, C.C., Knight, R., White, O., Huttenhower, C.: Assessment of variation in microbial community amplicon sequencing by the Microbiome Quality Control (MBQC) project consortium. *Nat. Biotechnol.* **35**(11), 1077–1086 (2017)
15. Tyx, R., Rivera, A., Zhao, N., Satten, G.: Comparing biases of extraction methods in mock community data (with and without a biological matrix) and in real samples (2020, in preparation)
16. van den Boogaart, K.G., Tolosana-Delgado, R.: “Compositions”: a unified R package to analyze compositional data. *Comput. Geosci.* **34**, 320–338 (2008)
17. Wang, Y.: Solving least squares or quadratic programming problems under equality/inequality constraints (2014)
18. Wang, Y.: CovTools: statistical tools for covariance analysis (2019)

Part IV

Bayesian Methods

Dirichlet-Multinomial Regression Models with Bayesian Variable Selection for Microbiome Data



Matthew D. Koslovsky and Marina Vannucci

1 Introduction

Human microbiome research aims to understand how microbiome communities interact with their host, respond to their environment, and influence disease [32]. High-throughput sequencing technologies have enabled researchers to characterize the composition of the microbiome by quantifying richness, diversity, and abundances. See [14] for a detailed review. However, complex environmental interactions with the microbiome challenge our understanding of community function and its impact on health [23]. Knowledge of the relations between microbial composition and other covariates may help researchers design tailored interventions to help maintain a healthy microbiome community [10, 33].

A popular approach for modeling the relation between microbial data and covariates is the Dirichlet-multinomial (DM) regression model, since it appropriately handles the compositional structure of microbiome data and accommodates overdispersion induced by sample heterogeneity and varying proportions among samples [3, 11–13, 28, 34]. To identify potential covariates, penalized likelihood methods have been developed to simultaneously estimate regression coefficients and perform selection [3, 30]. These models typically have relatively short computation times and demonstrate good predictive accuracy. However, it is challenging to incorporate known relations between covariates into these models due to the requirement of

M. D. Koslovsky
Colorado State University, Fort Collins, CO, USA
e-mail: Matt.Koslovsky@colostate.edu

M. Vannucci (✉)
Rice University, Houston, TX, USA
e-mail: marina@rice.edu

complex optimization routines [30]. Additionally, they do not accommodate model selection uncertainty while carrying out selection.

Alternatively, Bayesian variable selection methods are able to accommodate the complex high-dimensional data structures found in microbiome studies and fully account for model uncertainty over covariate selection. Commonly, spike-and-slab priors for regression coefficients are embedded into hierarchical Bayesian models to perform variable selection [8]. In this model formulation, regression coefficients' priors depend on latent inclusion indicators which determine a covariate's exclusion or inclusion in the model. Bayesian DM regression models with spike-and-slab priors were originally investigated by Wadsworth et al. [28] to identify KEGG orthology pathways associated with microbiome data. Through simulations, they demonstrate improved performance of their method on selecting covariates when compared to alternative methods, including the penalized likelihood approach of [3]. Recently, the work of Wadsworth et al. [28] was extended to accommodate phylogenetic structure between taxa and known and unknown graphical relations between covariates [11]. Additionally, researchers have leveraged data augmentation techniques to efficiently embed DM regression models into joint modeling frameworks, in order to investigate how the microbiome may mediate the relation between dietary factors and phenotypic responses, such as body mass index [12].

In an effort to make advanced Bayesian methods available to researchers studying the microbiome, we demonstrate how to apply the methods contained in `MicroBVS`, a comprehensive R package for identifying covariates associated with compositional data [11]. At the core of `MicroBVS` is a suite of Markov chain Monte Carlo (MCMC) algorithms that generate posterior samples of model parameters for inference. The MCMC algorithms are written in C++ to increase performance time and accessed through R wrapper functions using `Rcpp` and `RcppArmadillo` [5, 6]. The package includes various Bayesian variable selection methods for compositional data including Dirichlet-multinomial regression, Dirichlet-tree multinomial regression, and the joint modeling approach proposed in [12]. The package has built-in functionality to simulate data in user-specified research scenarios to assess selection performance and conduct sensitivity analyses. Additionally, various auxiliary R functions are incorporated to help researchers assess convergence, draw inference from the MCMC samples, and plot results. The package includes a vignette with worked examples using simulated data and access to open-source data used in our analyses.

In Sect. 2, we describe Dirichlet-multinomial (DM) and Dirichlet-tree multinomial (DTM) regression models with spike-and-slab priors and discuss alternative priors for inclusion indicators that accommodate known and unknown graphical structures between covariates. In Sect. 3, we perform a sensitivity and simulation study for Bayesian DM and DTM regression models and compare them to penalization approaches. Section 4 illustrates the application of the `MicroBVS` package to microbiome data collected in the Multi-omics Microbiome Study—Pregnancy Initiative and a benchmark dataset to investigate the relations between gut microbial taxa and dietary covariates. Section 5 provides concluding remarks.

2 Methods

2.1 Dirichlet-Multinomial Regression Models for Compositional Data

In this section, we introduce how to model compositional abundance data via a Dirichlet-multinomial (DM) regression framework and then demonstrate how to embed spike-and-slab priors for variable selection, similar to [28]. We first assume that taxa counts $y_i = (y_{i,1}, \dots, y_{i,K})$ follow a multinomial distribution

$$y_i \sim \text{Multinomial}(\dot{y}_i | p_i), \quad (1)$$

with $\dot{y}_i = \sum_{k=1}^K y_{i,k}$, and p_i defined on the K -dimensional simplex

$$S^{K-1} = \left\{ (p_{i,1}, \dots, p_{i,K}) : p_{i,k} \geq 0, \forall k, \sum_{k=1}^K p_{i,k} = 1 \right\}.$$

To account for overdispersion, we specify a conjugate prior on the taxa probabilities,

$$p_i \sim \text{Dirichlet}(\gamma_i), \quad (2)$$

with the K -dimensional vector $\gamma_i = (\gamma_{i,k} > 0, \forall k \in K)$, similar to [13] and [28]. Typically, the p_i are integrated out of the model for computational convenience, and the y_i are modeled with a $\text{Dirichlet-multinomial}(\gamma_i)$ [28]. To incorporate covariate effects into the model, we use a log-linear regression framework for the concentration parameters γ_i . Specifically, we set $\lambda_{i,k} = \log(\gamma_{i,k})$ and assume

$$\lambda_{i,k} = \alpha_k + \mathbf{x}'_i \boldsymbol{\varphi}_k, \quad (3)$$

where $\boldsymbol{\varphi}_k = (\varphi_{k1}, \dots, \varphi_{kP})'$ represents the covariates' potential relation with the k th compositional taxon, and α_k is a taxon-specific intercept term. Additionally, \mathbf{x}_i represents a P -dimensional vector of observed covariates for individual i , e.g., age, sex, medication use, and dietary factors. By exponentiating (3), we ensure positive hyperparameters for the Dirichlet distribution.

2.2 Variable Selection Priors

For DM regression models, the number of potential models to choose from when performing variable selection, 2^{PK} , grows quickly even for small covariate spaces. To induce sparsity in the model, we embed multivariate spike-and-slab priors for variable selection that identify covariates that are associated with each

compositional taxon [20, 24], as opposed to spike-and-slab constructions that select variables as relevant to either all or none of the responses [2]. We assume that the covariates' inclusion in the model is represented by a latent $K \times P$ -dimensional inclusion matrix ζ . As such, $\zeta_{kp} = 1$ indicates that covariate p is associated with compositional taxon k and 0 otherwise. The prior for φ_{kp} given ζ_{kp} follows a mixture of a normal distribution and a Dirac-delta function at zero, δ_0 , and is commonly referred to as the spike-and-slab prior. Specifically,

$$\varphi_{kp} | \zeta_{kp}, r_k^2 \sim \zeta_{kp} \cdot N(0, r_k^2) + (1 - \zeta_{kp}) \cdot \delta_0(\varphi_{kp}), \quad (4)$$

where r_k^2 is set large to impose a diffuse prior for the regression coefficients included in the model.

The DM model can incorporate different sparsity levels and can accommodate various structural relations between covariates through the specification of the prior probability of inclusion for each covariate, w_{kp} . Commonly, a beta-binomial distribution is assumed. With this prior, we let each ζ_{kp} follow a Bernoulli distribution

$$p(\zeta_{kp} | w_{kp}) = w_{kp}^{\zeta_{kp}} (1 - w_{kp})^{1 - \zeta_{kp}}$$

and further assume $w_{kp} \sim \text{Beta}(a, b)$. By integrating out w_{kp} , we obtain

$$p(\zeta_{kp}) = \frac{\text{Beta}(\zeta_{kp} + a, 1 - \zeta_{kp} + b)}{\text{Beta}(a, b)},$$

where the hyperparameters a and b can be set to impose different levels of sparsity in the model. In practice, the authors in [28] suggest using a weakly informative prior probability of inclusion by setting $a + b = 2$, where the prior expected mean value $m = a/(a + b)$. Thus, setting $a = 0.1$ and $b = 1.9$ reflects a prior belief that 5% of the covariates will be selected. A non-informative prior is assumed by setting $a = b = 1$ (i.e., $m = 0.50$). See [28] for a detailed sensitivity analysis regarding hyperparameter specification for DM regression models. To complete the model's specification, we assume that the intercept terms α_k follow a $N(0, \sigma_k^2)$, where σ_k^2 are set large to impose diffuse priors.

2.3 Network Priors

Under the beta-binomial prior, inclusion indicators are assumed independent. In other settings, researchers may be interested in incorporating prior information for the probability of inclusion of a covariate based on known relations with other covariates. For example, when covariates are chosen as gene expression levels, a network of covariate interactions may be known based on biological information [15, 25]. This graphical structure can be incorporated into the model via Markov

random field (MRF) priors, which are parameterized to increase a covariate's inclusion probability if neighboring covariates in the graph are included. MRFs are undirected graphical models for random variables whose distribution follows Markovian properties.

To use this information to help guide variable selection, the prior probability of inclusion for each covariate is set according to the given relations between covariates x . Specifically, we assume an MRF prior on ζ_k that increases the probability of inclusion for a covariate if covariates in its neighborhood in the graph are also included. Given the graph G , an adjacency matrix that represents the relations between covariates, the prior probability of inclusion for indicators ζ_k follows

$$p(\zeta_k|G) \propto \exp(a_G \mathbf{1}' \zeta_k + b_G \zeta'_k G \zeta_k),$$

where $\mathbf{1}$ is a P -dimensional vector of 1s and a_G and b_G control the global probability of inclusion and the influence of neighbors' inclusion on a covariate's inclusion, respectively. Previous studies have demonstrated how small increments in b_G can drastically increase the number of covariates included in the model [15, 25]. Li and Zhang [15] provide a detailed description of how to select a value for b_G . Note that if there is no structure between covariates, the prior probabilities of inclusion simplify to independent Bernoulli($\exp(a_G)/(1 + \exp(a_G))$).

2.3.1 Unknown G

When less is known about the relations between covariates, the network structure, G , can be inferred. Efficient sampling algorithms for learning the structure of high-dimensional data with Gaussian graphical models [29] have allowed researchers to embed them into Bayesian variable selection models that simultaneously perform variable selection while learning the relations between covariates [19].

Let $X \sim MVN(\mathbf{0}, \Omega)$, where $\Omega = \Sigma^{-1}$ is a $P \times P$ precision matrix. Following [29], we assume a hierarchical prior that models conditional dependence between covariates through edge detection in an undirected graph. Let graph G contain P nodes, corresponding to the set of potential covariates in the model. Let $g_{st} \in \{0, 1\}$ represent a latent inclusion indicator for an edge between nodes s and t , for $s < t$. The inclusion of edge g_{st} corresponds to $\omega_{st} \neq 0$, where ω_{st} , $1 \leq s < t \leq P$, are the off-diagonal elements of Ω . The prior distribution for Ω is the product of P exponential distributions for diagonal components and $P(P - 1)/2$ mixtures of normals for off-diagonal components of the precision matrix. Specifically,

$$p(\Omega|G, v_0, v_1, \theta) = \{C(G, v_0, v_1, \theta)\}^{-1} \prod_{s < t} N(\omega_{st}|0, v_{st}^2) \prod_s \text{Exp}(\omega_{ss}|\theta/2) I_{\{\Omega \in M^+\}},$$

where $\text{Exp}(\cdot|\theta/2)$ represents an exponential distribution with mean $2/\theta$, $C(G, v_0, v_1, \theta)$ is a normalizing constant, and $I_{\{\Omega \in M^+\}}$ is an indicator function that constrains Ω to be a symmetric positive definite matrix. Here, $v_{st}^2 = v_1$ if the

edge inclusion indicator $g_{st} = 1$, and $v_{st}^2 = v_0$ if $g_{st} = 0$. In practice, $v_0 > 0$ is set small to concentrate ω_{st} around zero for excluded edges, and $v_1 > 0$ is set large so that ω_{st} is freely estimated via a diffuse prior for included edges. The prior for the edge inclusion indicator g_{st} follows

$$p(G, v_0, v_1, \theta, \pi) = \{C(v_0, v_1, \theta, \pi)\}^{-1} C(G, v_0, v_1, \theta) \prod_{s < t} \left\{ \pi^{g_{st}} (1 - \pi)^{1-g_{st}} \right\},$$

where $C(v_0, v_1, \theta, \pi)$ is a normalizing constant and π represents the prior probability of inclusion for an edge. Following the recommendations of [29], the specification of π should reflect prior belief in the sparsity of the graph, and θ is typically set to one. The latter implies a relatively vague prior for ω_{ss} , since the data are usually standardized prior to analysis. See [29] for more details regarding prior specification.

2.4 Dirichlet-Tree Multinomial Models

In this section, we describe Bayesian variable selection for Dirichlet-tree multinomial regression models, similar to [11]. The DM model described in Sect. 2.1 assumes that counts are negatively correlated. Alternatively, the Dirichlet-tree multinomial model (DTM) inherits the DM's ability to handle overdispersed data, can model general correlation structures between counts, and can naturally incorporate structural information [4, 17]. In microbiome research, this allows us to model evolutionary relations among taxa represented by a phylogenetic tree [11, 26, 27, 30].

To accommodate a tree-like structure among counts, the multinomial distribution is deconstructed into the product of multinomial distributions for each of the subtrees in the tree, and the conjugate Dirichlet-tree prior is assumed [4]. Specifically, let tree T have K leaf nodes and V internal nodes. Let C_v represent the set of child nodes for each individual node $v \in V$. For each subject, the branch probability between parent node v and child node c is represented as $p_{i,vc}$, where $\sum_{c=1}^{|C_v|} p_{i,vc} = 1$ and $|C_v|$ is the number of child nodes of v . Under this parameterization, we assume that $y_{i,v} = (y_{i,v1}, \dots, y_{i,vC})'$ follows a $\text{Multinomial}(y_{i,v}, p_{i,v})$, where $p_{i,v} = \{p_{i,vc}, c \in C_v\}$. We assume a $\text{Dirichlet}(\gamma_{i,v})$ prior for each $p_{i,v}$, where $\gamma_{i,v} = (\gamma_{i,vc} > 0, \forall c \in C_v)$. Integrating the $p_{i,v}$ out, we model $y_{i,v}$ with a Dirichlet-multinomial($\gamma_{i,v}$) and take the product of the v Dirichlet-multinomial models for each sub-tree, to obtain the Dirichlet-tree multinomial (DTM) distribution as

$$p(y_i | \gamma_i, v \in V) = \prod_{v \in V} \frac{\Gamma(\sum_{c \in C_v} y_{i,vc} + 1) \Gamma(\sum_{c \in C_v} \gamma_{i,vc})}{\Gamma(\sum_{c \in C_v} y_{i,vc} + \sum_{c \in C_v} \gamma_{i,vc})} \times \prod_{c \in C_v} \frac{\Gamma(y_{i,vc} + \gamma_{i,vc})}{\Gamma(y_{i,vc} + 1) \Gamma(\gamma_{i,vc})},$$

where Γ represents the gamma function. The generalized DM model and the DM model are special cases of the DTM class of models [30]. Specifically, the

generalized DM model can be represented as a DTM with a binary cascading tree (i.e., at each level of the tree, the rightmost branch splits into two), and the DM can be represented with a tree containing only one root node and K leaf nodes.

Similar to Eq. (3), covariate effects can be incorporated into the model using a log-linear regression framework. Specifically, we set $\lambda_{i,vc} = \log(\gamma_{i,vc})$ and assume

$$\lambda_{i,vc} = \alpha_{vc} + \mathbf{x}'_i \boldsymbol{\varphi}_{vcP},$$

where $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,P})'$ represents a set of measurements on P covariates and $\boldsymbol{\varphi}_{vc} = (\varphi_{vc1}, \dots, \varphi_{vcP})'$. We assume that the intercept terms α_{vc} follow a $N(0, \sigma_{vc}^2)$, where σ_{vc}^2 are set large to impose vague priors on α_{vc} . Similar prior specifications for variable selection presented in Sect. 2.2 can be applied to each of the DM components of this model.

2.5 Posterior Inference

In Bayesian inference, the posterior distribution is proportional to the product of the likelihood of the data and the prior distributions for the parameters. For both DTM and DM models, researchers have implemented Metropolis–Hastings algorithms within a Gibbs sampler for inference [11, 28]. Since the DTM model is a generalization of the DM model, we present a general MCMC algorithm in the context of DTM models. Assuming a beta-binomial prior probability of inclusion, the parameter space is described as $\Phi = \{\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}\}$, and the posterior distribution is

$$p(\Phi|\mathbf{Y}, \mathbf{x}) \propto f(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}, \mathbf{x}) p(\boldsymbol{\alpha}) p(\boldsymbol{\varphi}|\boldsymbol{\zeta}) p(\boldsymbol{\zeta}).$$

We use a two-step update approach to sample regression coefficients and inclusion indicators for covariates, following [21].

A generic iteration of the MCMC algorithm is described as follows:

- Update each α_{vc} —Metropolis step with random walk proposal from $\alpha'_{vc} \sim N(\alpha_{vc}, 0.50)$. Accept proposal with probability

$$\min \left\{ \frac{f(\mathbf{Y}|\boldsymbol{\alpha}', \boldsymbol{\varphi}, \boldsymbol{\zeta}, \mathbf{x}) p(\alpha'_{vc})}{f(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}, \mathbf{x}) p(\alpha_{vc})}, 1 \right\}.$$

- Jointly update a ζ_{vcP} and $\boldsymbol{\varphi}_{vcP}$

- *Between-Model Step*: Randomly select a ζ_{vcP} term.

Add: If the covariate is currently excluded ($\zeta_{vcP} = 0$), change it to $\zeta'_{vcP} = 1$. Then, sample a $\boldsymbol{\varphi}'_{vcP} \sim N(\boldsymbol{\varphi}_{vcP}, 0.50)$. Accept proposal with probability

$$\min \left\{ \frac{f(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}', \boldsymbol{\zeta}', \mathbf{x}) p(\boldsymbol{\varphi}'_{vcp}|\zeta'_{vcp}) p(\zeta'_{vc})}{f(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}, \mathbf{x}) p(\boldsymbol{\varphi}_{vcp}|\zeta_{vcp}) p(\zeta_{vc})}, 1 \right\}.$$

Delete: If the covariate is currently included ($\zeta_{vcp} = 1$), change it to $\zeta'_{vcp} = 0$ and set $\boldsymbol{\varphi}'_{vcp} = 0$. Accept proposal with probability

$$\min \left\{ \frac{f(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}', \boldsymbol{\zeta}', \mathbf{x}) p(\zeta'_{vc})}{f(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}, \mathbf{x}) p(\boldsymbol{\varphi}_{vcp}|\zeta_{vcp}) p(\zeta_{vc})}, 1 \right\}.$$

- *Within-Model Step:* Propose a $\boldsymbol{\varphi}'_{jp} \sim N(\boldsymbol{\varphi}_{jp}, 0.50)$ for each covariate currently selected in the model ($\zeta_{vcp} = 1$). Accept each proposal with probability

$$\min \left\{ \frac{p(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}', \boldsymbol{\zeta}, \mathbf{x}) p(\boldsymbol{\varphi}'_{vcp}|\zeta_{vcp})}{p(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}, \mathbf{x}) p(\boldsymbol{\varphi}_{vcp}|\zeta_{vcp})}, 1 \right\}.$$

To include a known graphical structure and impose an MRF prior for selection, the algorithm simply replaces $p(\boldsymbol{\zeta})$ with $p(\boldsymbol{\zeta}|G)$. If the relational structure between the covariates is unknown, the posterior distribution of the model is redefined as

$$p(\Phi|\mathbf{Y}, \mathbf{X}) \propto f(\mathbf{Y}|\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}, \mathbf{X}) f(\mathbf{X}|\boldsymbol{\Omega}) p(\boldsymbol{\alpha}) p(\boldsymbol{\Omega}|G) p(\boldsymbol{\varphi}|\boldsymbol{\zeta}) p(\boldsymbol{\zeta}|G) p(G),$$

where $\Phi = \{\boldsymbol{\alpha}, \boldsymbol{\varphi}, \boldsymbol{\zeta}, \boldsymbol{\Omega}, G\}$. Note that this parameterization treats the covariates \mathbf{X} as random and not fixed. For implementation, the MCMC algorithm requires two additional steps to simultaneously learn the graphical relations. We update $\boldsymbol{\Omega}$ and G following the approach outlined in [29].

For implementation, the algorithms are initiated at a set of arbitrary parameter values and then used to generate samples of the posterior distribution. After burn-in, the remaining samples are used for inference. To determine inclusion in the model, the marginal posterior probability of inclusion (MPPI) for each of the covariates is determined by taking the average of their respective inclusion indicator's MCMC samples. Note that a covariate has a unique inclusion indicator for each of the taxon. Commonly, variables are included in the model if their MPPI ≥ 0.50 [1]. Alternatively, the authors in [18] propose using a threshold based on a Bayesian false discovery rate (BFDR) to control for multiplicity.

3 Simulated Data

In this section, we demonstrate the selection performance for the DM and DTM models using simulated data. For the DM models, we compared the performances using different variable selection priors, i.e., a beta-binomial prior, an MRF prior

with fixed graphical structure (i.e., G set to the truth and G learned a priori), and an MRF prior with unknown graphical structure.

For variable selection, all models were assessed on the basis of sensitivity (1—false negative rate), specificity (1—false positive rate), and Matthew’s correlation coefficient (MCC) (a measure of overall selection accuracy). These are defined as

$$\text{Sensitivity} = \frac{\text{TP}}{\text{FN} + \text{TP}}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}},$$

where TN, TP, FN, and FP represent the true negatives, true positives, false negatives, and false positives, respectively. Covariates were determined to be associated with the compositional and response data, respectively, if their MPPI ≥ 0.50 [1]. Results we report below were obtained by averaging over 30 replicated datasets.

3.1 Simulation Study for DM Regression Models

Similar to simulation schemes adopted by [3, 12, 28], we simulated $N = 100$ subjects with $P = 30$ covariates and $K = 75$ compositional taxa. Covariates \mathbf{x} were simulated from a $N_P(\mathbf{0}, \Sigma)$, where Σ was set to a block diagonal matrix with one along the diagonal and three 5×5 exchangeable covariance structures (for the first 15 covariates) with $\sigma_{ij} = 0.7, 0.5$, and 0.3 , respectively. In each of the replicate datasets, we randomly selected 25 of the 2250 covariate–taxon combinations to be associated with the compositional data. Corresponding regression coefficients φ were randomly sampled from $\pm[0.75, 1.25]$. Intercept terms α were simulated from a Uniform $[-2.3, 2.3]$. The compositional data \mathbf{Y} were sampled from a Multinomial($\dot{\mathbf{y}}_i, p_i^*$), where $\dot{\mathbf{y}}_i \sim \text{Uniform}[5,000, 10,000]$ and $p_i^* \sim \text{Dirichlet}(\boldsymbol{\gamma}_i^*)$, where $\boldsymbol{\gamma}_i^* = (\gamma_{i,1}^*, \gamma_{i,2}^*, \dots, \gamma_{i,K}^*)$. We let $\gamma_{i,k}^* = \frac{\gamma_{i,k}}{\sum_{k=1}^K \gamma_{i,k}} \frac{1-d}{d}$, $k = 1, \dots, K$, where $\gamma_{i,k}$ was determined using Eq. (3), and d serves as an overdispersion parameter which was set at 0.01. As a result, the data-generating model differs from our model assumptions.

When running the MCMC algorithm, we set hyperparameters $a = 1$ and $b = 9$ for the beta-binomial prior and $a_G = \log(0.1/0.9)$ for the MRF prior, representing a prior expectation of 10% of the total number of covariates included in both models. For the MRF prior with known graphical structure, we set $b_G = 0.2$ and the graph G equal to a $P \times P$ -dimensional block diagonal matrix, with $3, 5 \times 5$ blocks of 1s for the first 15 elements. Additionally, we set G equal to the graphical structure learned

Table 1 Simulation results for the DM regression model with various inclusion indicator prior assumptions. # **Selected**—the number of selected covariates and **MCC**—Matthew’s correlation coefficient. Results are presented as mean (SD) over 30 replicate datasets

Prior	# Selected	Sensitivity	Specificity	MCC
beta-binomial	24.3 (3.2)	0.904 (0.092)	0.999 (0.001)	0.917 (0.062)
MRF fixed G -true	56.6 (13.8)	0.987 (0.028)	0.986 (0.006)	0.665 (0.083)
MRF fixed G -learned	43.8 (12.4)	0.975 (0.032)	0.991 (0.006)	0.748 (0.086)
MRF unknown G	42.6 (8.5)	0.979 (0.029)	0.992 (0.003)	0.766 (0.073)

using [29]. For the MRF prior with unknown graphical structure, we set $b_G = 0.2$, $v_0 = 0.01$, $v_1 = 10$, $\lambda = 1$, and $\pi = 2/(P - 1)$, similar to [29]. Simulations were run for 10,000 iterations and thinned to every 10th iteration. This resulted in 1,000 iterations, of which the first 500 iterations were treated as burn-in and the remaining 500 used for inference. Each run was initiated with $\zeta_{pk} = 0$ and α_k sampled from a standard normal distribution.

Results are found in Table 1. Overall, the DM model with MRF prior and fixed graphical structure among covariates had the highest number of selected covariates on average. These results were expected since the baseline prior probability of inclusion using the MRF (a_G) was set to impose a 10% prior probability of inclusion, similar to the beta-binomial model, and any graphical structure (known or unknown) would only increase the probability of inclusion in the model. As a result, the MRF with G fixed to the truth had the highest sensitivity overall. However, since it typically overselected, it achieved the lowest specificity and MCC as well. Overall, the DM with a beta-binomial prior had the highest MCC ($\sim 92\%$). Lastly, we observed a marginal improvement in selection performance when learning the graphical structure simultaneously in the model versus a priori. It is important to note that the MRF model with unknown graphical structure had similar performance to the MRF with known graphical structure while additionally providing inference on the relations among covariates.

3.2 DM Sensitivity Analysis

To assess the model’s sensitivity to hyperparameter settings, we set each of the hyperparameters to default values and then evaluated the effect of manipulating each term on selection performance. We investigated the model’s sensitivity to specification of the beta-binomial prior hyperparameters a and b , MRF prior hyperparameters a_G and b_G , and hyperparameters associated with the Gaussian graphical models, v_0 , v_1 , and π . For the default parameterization, we set the hyperparameters for the beta-binomial prior inclusion indicators to $a = 1$ and $b = 9$. For the MRF priors, we set the hyperparameters $a_G = \log(0.1/0.9)$ and $b_G = 0.2$. The default values for the Gaussian graphical model hyperparameters were $v_0 = 0.01$, $v_1 = 10$, and $\pi = 2/(P - 1)$. We ran our MCMC algorithm on

Table 2 Sensitivity results for the beta-binomial and MRF prior probability of inclusion parameters b and b_G , respectively, the exclusion variance for graphical edge selection v_0 , and the prior probability of edge inclusion π . # Selected—the number of selected covariates. MCC—Matthew’s correlation coefficient. Results are presented as mean (SD) over 30 replicate datasets.

Prior		$b = 1$	$b = 99$
beta-binomial	# Selected	37.0 (9.4)	21.5 (2.5)
	Sensitivity	0.97 (0.04)	0.83 (0.12)
	Specificity	0.99 (0.00)	1.00 (0.00)
	MCC	0.81 (0.09)	0.89 (0.08)
	$b_G = 0.05$	$b_G = 0.5$	
MRF fixed G	# Selected	41.2 (10.9)	893.6 (72.4)
	Sensitivity	0.97 (0.04)	1.00 (0.00)
	Specificity	0.99 (0.00)	0.61 (0.03)
	MCC	0.77 (0.09)	0.13 (0.01)
	$v_0 = 0.001$	$v_0 = 0.1$	
MRF unknown G	# Selected	43.9 (10.7)	41.9 (9.8)
	Sensitivity	0.98 (0.03)	0.97 (0.03)
	Specificity	0.99 (0.00)	0.99 (0.00)
	MCC	0.75 (0.09)	0.76 (0.08)
	$\pi = 0.02$	$\pi = 0.5$	
MRF unknown G	# Selected	42.5 (11.6)	47.1 (12.3)
	Sensitivity	0.98 (0.04)	0.98 (0.03)
	Specificity	0.99 (0.01)	0.99 (0.01)
	MCC	0.76 (0.09)	0.72 (0.08)

the 30 replicated datasets generated in the simulation study, using 10,000 iterations, treating the first 5,000 iterations as burn-in, and thinning to every 10th iteration.

The results of the sensitivity analysis are presented in Table 2. As expected, we found that increasing (decreasing) b in the beta-binomial prior reduced (increased) the number of covariates selected in the model. Here, we observed a positive relation between sensitivity and the prior probability of inclusion. However, since the model overselected covariates with smaller b values, the specificity diminished as a result. Using an MRF prior with a fixed underlying graphical structure, we found that as b_G increased, so did the number of selected covariates on average. In our analysis, the models seemed to experience a phase transition, in which the number of covariates selected in the model dramatically increased, for $b_G = 0.5$. See [15] for recommendations on selecting the appropriate b_G in practice. With unknown graphical structure, we found marginal differences in results relative to changes in v_0 and π .

3.3 Simulation Study for DTM Regression Models

For the DTM model, we compared selection performances to the penalized DTM approach of [30]. We simulated $N = 100$ subjects with $P = 75$ covariates and $K = 30$ compositional taxa. Covariates x were simulated from a $N_P(\mathbf{0}, \Sigma)$, where

$\sigma_{ij} = \omega^{|i-j|}$ and $\omega = 0.3$. In each of the replicate datasets, we randomly selected 15 of the 4,350 covariate–branch combinations to be associated with the compositional data. Corresponding regression coefficients φ were randomly sampled from $\pm[0.75, 1.50]$. Intercept terms α were simulated from a Uniform $[-1.3, 1.3]$. The multivariate count data \mathbf{Y} were sampled from a DTM regression model with total counts for each individual uniformly distributed between 7,500 and 10,000. For each dataset, we simulated a random tree using sequential binary separation [7], in which the parent node and subsequent internal nodes are split into two branches until the total number of leaf nodes K is obtained.

We chose a beta-binomial inclusion prior and set $a = 1$ while varying b as $b = 1, 9$, and 99 , to investigate the model’s sensitivity to hyperparameter specification. The MCMC algorithms were run for 40,000 iterations, treating the first 20,000 as burn-in and thinning to every 10th iteration. For the penalized approach of [30], it is necessary to choose tuning parameters γ and λ , which control the sparsity of the model. When $\gamma = 0$ and $\gamma = 1$, the model generates the lasso and group lasso estimate, respectively. Following the recommendations of [30], we set $\gamma = \{0.0, 0.25, 0.5, 1.0\}$ and fit the model over a grid of λ values. The best model for each γ was then chosen by minimizing the Bayesian information criterion [22].

Similar to the DM model, we found that the DTM was sensitive to the prior probability of inclusion (Table 3). Specifically, as b increased (decreased), the number of covariate–branch association decreased (increased), as expected. We found that the model with $b = 9$ had the best selection performance overall ($MCC = 0.544$), and the non-informative model (i.e., $a = b = 1$) showed the worst performance overall ($MCC = 0.219$). All prior specifications achieved a relatively high specificity (>0.97). Similar specificity results were found with the penalized approach (Table 4). However, the penalized approach, regardless of tuning parameter γ , had extremely low sensitivity, resulting in low MCC values as well. When $\gamma = 1$, the penalized model did not select any covariate–branch terms (results not shown).

Table 3 Simulation results for the Bayesian variable selection method for DTM regression models at various prior probabilities of inclusion

Prior	# Selected	Sensitivity	Specificity	MCC
$a = 1$ and $b = 1$	135.0 (42.1)	0.642 (0.184)	0.971 (0.010)	0.219 (0.085)
$a = 1$ and $b = 9$	15.5 (5.2)	0.564 (0.239)	0.998 (0.001)	0.544 (0.202)
$a = 1$ and $b = 99$	5.2 (2.6)	0.293 (0.145)	1.00 (0.00)	0.491 (0.156)

Table 4 Simulation results for the penalized DTM regression approach of [30]. For each γ , the optimal model is chosen over a grid of λ values using the Bayesian information criterion

γ	# Selected	Sensitivity	Specificity	MCC
0.0	47.5 (36.3)	0.122 (0.172)	0.989 (0.008)	0.071 (0.098)
0.25	28.3 (25.1)	0.107 (0.173)	0.994 (0.006)	0.090 (0.142)
0.50	17.3 (21.7)	0.071 (0.139)	0.996 (0.005)	0.076 (0.118)

Table 5 Sensitivity results for high and low count associations with the Bayesian beta-binomial ($a = 1$ and $b = 9$) and Penalized DTM regression models. **# Selected**—the number of selected covariates and **MCC**—Matthew’s correlation coefficient. Results are presented as mean (SD) over 30 replicate datasets

Branch count	Model	# Selected	Sensitivity	Specificity	MCC
High	Bayesian	19.2 (3.5)	0.507 (0.106)	0.998 (0.001)	0.575 (0.097)
	Penalized	49.6 (35.8)	0.800 (0.089)	0.993 (0.008)	0.629 (0.135)
Low	Bayesian	17.7 (5.4)	0.466 (0.165)	0.999 (0.001)	0.546 (0.131)
	Penalized	255.1 (320.1)	0.542 (0.220)	0.944 (0.074)	0.270 (0.189)

3.4 DTM Sensitivity Analysis

In this sensitivity analysis, we investigate how selection performance is affected by branch count. Specifically, we simulated data similar to Sect. 3.3, with the exception that we targeted high (upper quartile) and low (lower 50th percentile) branch count regions in the tree when setting the associated terms. In the first (second) setting, we activated 25 terms across 5 high (low) count branches. We applied the Bayesian and penalized approaches used in the simulation study in this analysis and present results for the best performing parameterizations. For the Bayesian approach, we assumed a beta-binomial prior for inclusion indicators, ($a = 1$ and $b = 9$), and for the penalized approach, we set $\gamma = 0.50$.

The results of our sensitivity analysis are presented in Table 5. Here, we found that the Bayesian model was quite robust to branch counts. In both the high and the low settings, it generated selection performance results similar to the simulation study ($MCC \sim 0.55$). The penalized method showed the best performance overall when the covariates were associated with high branch counts ($MCC = 0.63$). However, in the low branch count setting, it over-selected, which greatly reduced its overall performance. Thus, in practice, the Bayesian method may be preferred in more sparse settings, whereas the penalized approach may be better suited for studies with higher numbers of taxa reads.

4 Applications

In this section, we apply the DM and DTM Bayesian variable selection methods to data collected in two microbiome studies, in order to demonstrate how to implement the MCMC algorithms provided in MicroBVS and how to draw inference on the results. First, we apply the DM regression model with beta-binomial and MRF priors for inclusion indicators to open-source data collected in the Multi-omics Microbiome Study—Pregnancy Initiative (MOMS-PI) [9]. This study was funded by the NIH Roadmap Human Microbiome Project with the aim of understanding the relations between the microbiome and pregnancy-related health outcomes. Then,

we demonstrate the functionality of the DTM regression model by applying it to a benchmark dataset collected to study the relation between the dietary intake and the human gut microbiome [31]. The data used in this analysis consist of 28 genera-level OTU counts obtained from 16S rRNA sequencing and a corresponding set of 97 dietary intake covariates derived from diet information collected using a food frequency questionnaire on 98 subjects.

4.1 Multi-omics Microbiome Study—Pregnancy Initiative (MOMS-PI)

To demonstrate the application of the DM regression models with various inclusion indicator priors, we use the open-source data collected in the Multi-omics Microbiome Study—Pregnancy Initiative (MOMS-PI). Data were obtained from the HMP2Data package in R, which contains observations on 596 subjects. Women enrolled in the study provided microbiome samples from the mouth, skin, vagina, and rectum longitudinally. We investigated relations between the vaginal microbiome and cytokine abundances, which help regulate the composition of the vaginal microbiome. For this analysis, we used baseline measures on 225 subjects with accompanying cytokine abundances. The dataset is available as part of the MicroBVS R package [11]. To install the package, follow the instructions in the README found at <http://github.com/mkoslovsky/MicroBVS>. Once installed, load the package, as well as the abundance, cytokine, and taxonomic data, into the R environment by running:

```
329 library(MicroBVS)
330 data("momspi16S")
331 data("momspiCyto")
332 data("momspi16S_tax")
```

We further limited analyses to only those taxa identified in at least 10% of participants (i.e., 123 taxa), to reduce the number of spurious relationships detected. We also standardized the cytokine values before analysis. When running the model with an MRF prior with an unknown graphical structure, cytokine abundances were log transformed and centered. Prior to transformation, cytokine values ≤ 0 were replaced with relatively small pseudovalues.

To fit the DM regression model with a non-informative beta-binomial prior for inclusion indicators, simply run

```
333 model1 <- DMbvs_R(iterations = 50000, thin = 10,
334                      z = momspi16S, x = momspiCyto,
335                      prior = "BB", a = 1, b = 1, seed = 1)
```

For the results given below, we ran the model for 50,000 iterations, thinning to every 10th and setting the initial seed at 1 for reproducibility. To extract the results from the DMbvs_R object, use the selected_DM() function as follows:

```
336 out <- selected_DM( model1, threshold = 0.5, burnin =
  2500)
```

The `out` object contains a # Selected covariates \times 2-dimensional matrix of associations, where the first (second) column represents the row (column) of the corresponding `momspiCyto` term selected using a burn-in of 2500 iterations and a marginal posterior probability of inclusion threshold of ≥ 0.5 , following the median model approach [1]. Additionally, the `out` object contains the MPPIs for all of the corresponding cytokine–taxon associations. Figure 1 presents a plot of the MPPIs for each covariate–taxon pair, and Fig. 2 is a heatmap of identified associations’ regression coefficients. For this analysis, the model selected 43 covariate–taxon associations.

Next, we ran the DM regression model with an MRF prior with an unknown graphical structure as

```
337 model2 <- DMbvs_R(iterations = 50000, thin = 10,
338                     z = momspi16S, x = momspiCyto,
339                     prior = "MRF_unknown",
340                     a_G = 0, b_G = 0.2, v0 = 0.01, v1 = 10,
341                     pie = 2/(ncol(momspiCyto)-1), lambda = 1)
```

We assumed the baseline prior probability of inclusion, a_G , equal to zero (analogous to the non-informative beta-binomial prior), and the rest of the hyperparameters were set similarly to our simulation study. Results from `model2` can be extracted using the `selected_DM` function as above. To extract the learned graphical

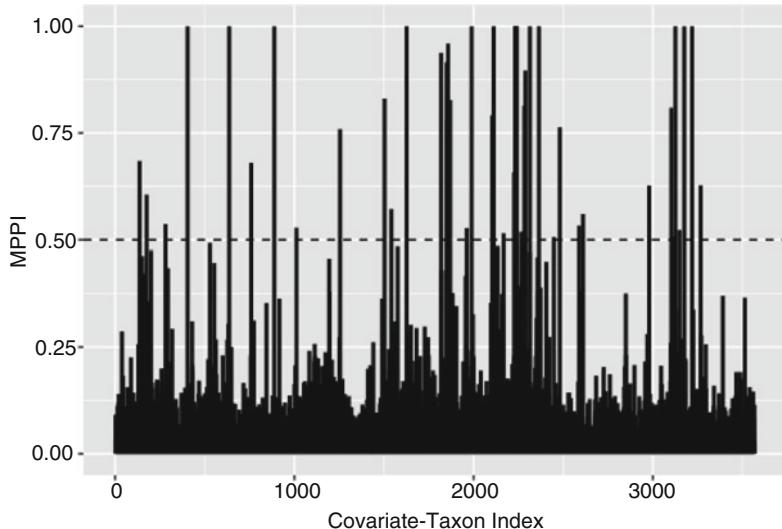


Fig. 1 MOMS-PI study: resulting marginal posterior probability of inclusion from DM regression model with beta-binomial priors for inclusion indicators. MPPI threshold of 0.50 indicated with dotted line

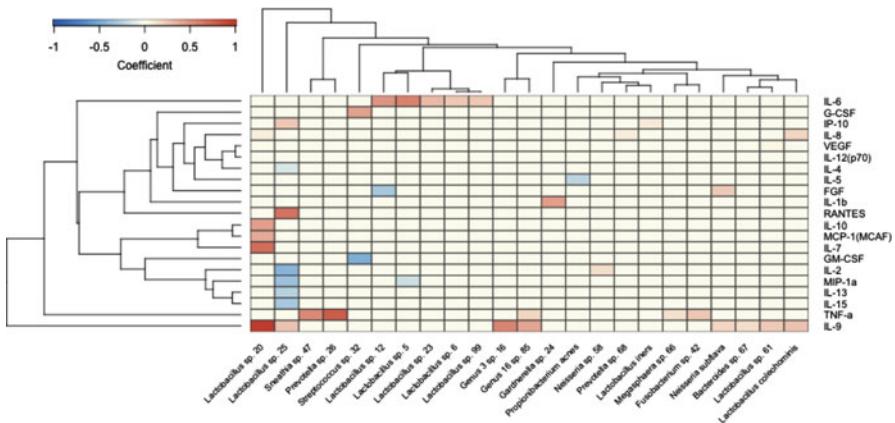


Fig. 2 MOMS-PI study: heatmap of cytokine-taxon associations identified with DM regression model with beta-binomial priors. Taxa are indexed by genus and species

structure in the cytokine data, additionally set the argument $G = T$. This generates an additional `estimated_G` element for the `selected_DM` object, which is a $\# \text{ cytokines} \times \# \text{ cytokines}$ -dimensional adjacency matrix. A network plot of the learned structure is presented in Fig. 3. With the MRF prior, the number of included covariate–taxon associations increased to 64, as expected from the simulation study. A plot of the MMPIs for `model12` is presented in Fig. 4, and the corresponding heatmap of identified association is presented in Fig. 5. To fit the DM regression model with fixed graphical structure between covariates, set the `DMbvs_R` function argument `prior = "MRF_fixed"` and `G` equal to an adjacency matrix representing the assumed graphical structure. Additional examples on simulated data can be found in the vignette provided with the `MicroBVS` package.

4.2 Gut Microbiome Study

In this section, we demonstrate how to apply the DTM Bayesian variable selection method to a benchmark dataset collected to study the relation between the dietary intake and the human gut microbiome [31]. Previously, Wang and Zhao [30] proposed a penalized DTM regression model to identify dietary intake covariates associated with genus-level operational taxonomic units (OTUs) on a subset of these data. We illustrate the Bayesian DTM model on the same subset. To load the necessary R packages and data into the R environment, run

```
342 library(MicroBVS)
343 library(phyloseq)
344 data("Gut_micro")
```

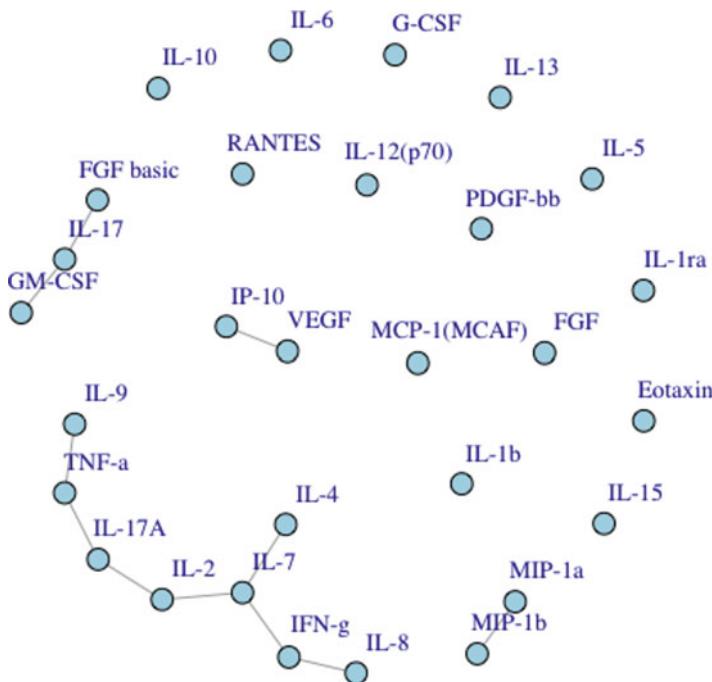


Fig. 3 MOMS-PI study: learned graphical structure of cytokine data

```
345 data("Gut_dietary")
346 data("tree")
```

The phylogenetic tree used in this example is presented in Fig. 6. We assumed a non-informative beta-binomial prior for inclusion indicators ($a = b = 1$). The MCMC algorithm was run for 150,000 iterations thinning to every 100th sample. After a burn-in of 750 samples, inference was drawn from the remaining 750.

```
347 model_gut <- DTMBvs_R( iterations = 150000, thin =
  100, tree = tree, Y = Gut_micro, X = Gut_dietary,
348 prior = "BB", seed = 1)
```

In this example, we used a Bayesian false discovery rate of 0.01 to determine a covariate's inclusion in the model. To identify the corresponding MPPI threshold for inclusion, run the `selected_DTM` function to obtain the matrix of MPPIs. Then, run the `bfdr` function at the prespecified error level, i.e., 0.01 in this application. Next, run the `selected_DTM` function with the BFDR threshold, $\text{MPPI} \geq 0.89$ in this example. To label the covariates, we supplied the column names for the `Gut_dietary` matrix. While not shown here, the function also has an argument for edge labels (`edge_lab`) to help with inference. See the vignette for more details.

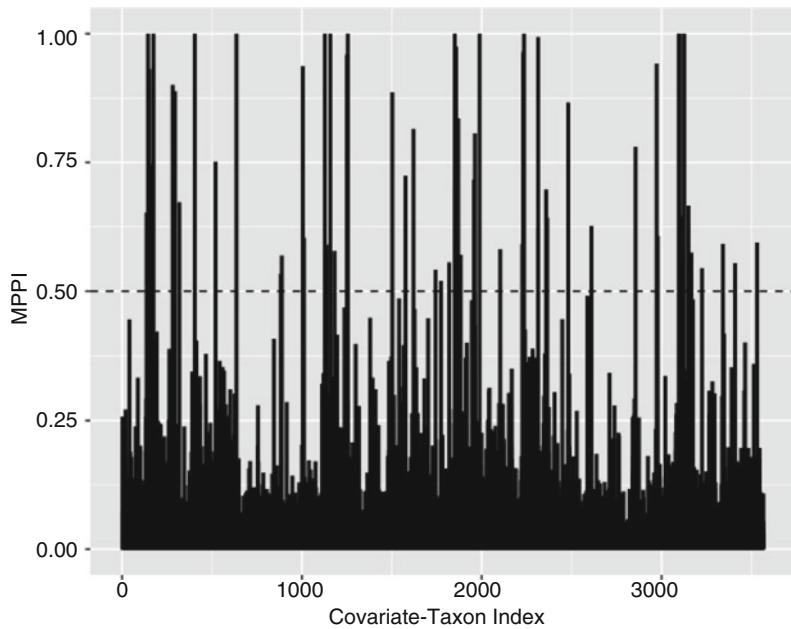


Fig. 4 MOMS-PI study: resulting marginal posterior probability of inclusion for results from DM regression model with MRF prior for inclusion indicators. MPPI threshold of 0.50 indicated with dotted line

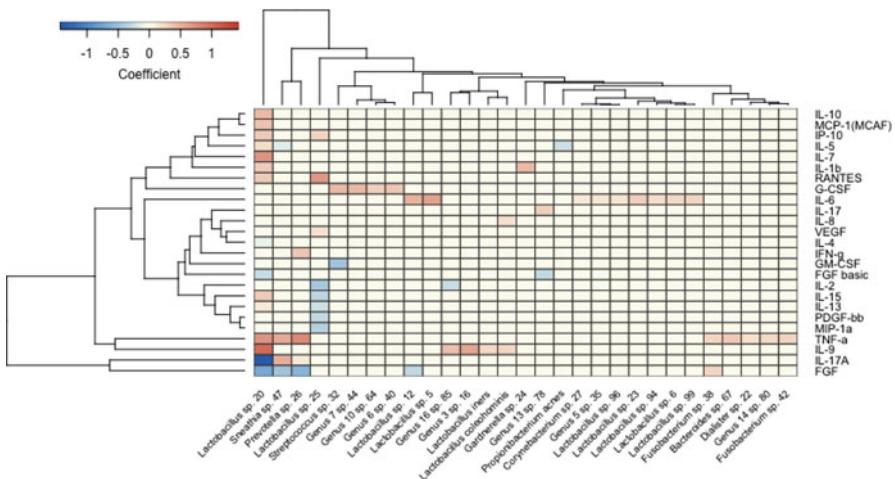


Fig. 5 MOMS-PI study: heatmap of cytokine-taxon associations identified with DM regression model with MRF priors. Taxa are indexed by genus and species

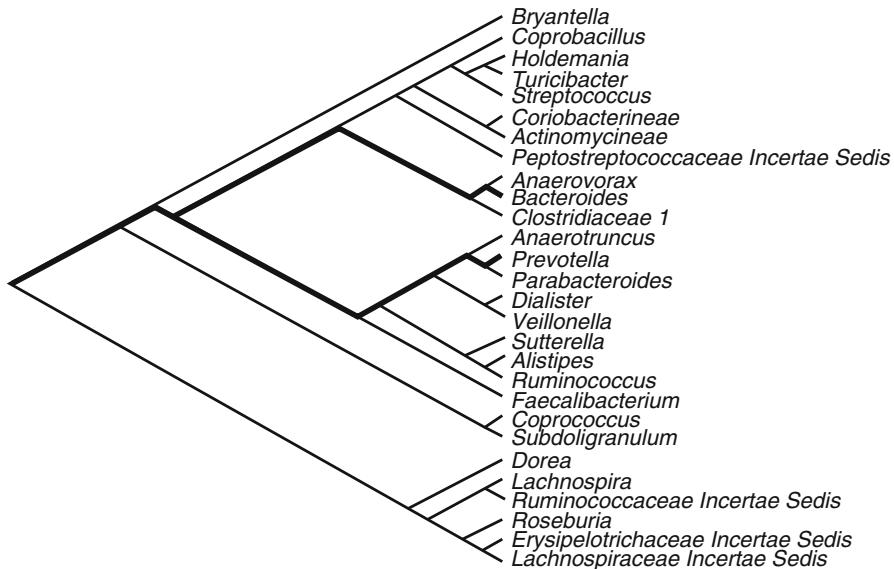


Fig. 6 Gut microbiome study: phylogenetic tree for identifying dietary intake covariates associated with genus-level OTUs in a Dirichlet-tree multinomial model regression

```

349 MPPI <- selected_DTM( model_gut, burnin = 750 )$mppi_
      zeta
350 bfdr_fit <- bfdr( MPPI, threshold = 0.01 )
351 out <- selected_DTM( model_gut, burnin = 750,
352                         threshold = bfdr_fit$threshold,
353                         cov_lab = colnames( Gut_dietary ) )

```

For inference, we are interested in the dietary covariates associated with branches along the path from a particular taxon to the root node. For demonstration, we focus on two genera researchers that have previously targeted in these data [11, 30], *Bacteroides* and *Prevotella*. To find the unique covariates associated with the branches corresponding to *Bacteroides*, run the `branch_covariates` function as below:

```

354 bact_cov <- branch_covariates( tree = tree, dtm_obj =
355 model_gut, covariate_name = colnames( Gut_dietary ),
356 branch_name = "Bacteroides", threshold = bfdr_fit\$
      threshold )

```

This function generates a vector of the unique covariates associated with a given taxon. Note that the `branch_name` provided must match an element in the `covariate_name` vector. In Table 6, we present the dietary intake covariates selected by the Bayesian DTM regression model for *Bacteroides* and *Prevotella*.

Table 6 Gut microbiome study: dietary factors identified as associated with *Bacteroides* and *Prevotella* using the DTM model with Bayesian variable selection

<i>Bacteroides</i>	<i>Prevotella</i>
Protein	Saturated fat
Saturated fat	Palmitic fatty acid
Palmitic fatty acid	Stearic fatty acid
Stearic fatty acid	Natural food folate
Natural food folate	Retinol equivalents of vitamin A
Vitamin E, food fortification	Vitamin E, food fortification
Maltose	Palmitelaidic trans fatty acid
Total trans	c9,t11 conjug diene isomer 18:2 Linoleic
Isoleucine	Total trans
Lysine	Isoleucine
Phenylalanine	Arginine
Histidine	Serine
Serine	Delphinidin, anthocyanidin
Naringenin, flavanone	Petunidin, anthocyanidin
Delphinidin, anthocyanidin	Proanthocyanidin, trimers
Petunidin, anthocyanidin	
Proanthocyanidin, trimers	
Proanthocyanidin, polymers	

5 Conclusion

In this chapter, we have detailed the use of Dirichlet-multinomial-based approaches with Bayesian variable selection for microbiome studies. We have explored various priors for inclusion indicators using the DM regression model and additionally demonstrated how to incorporate phylogenetic structure into the analysis using DTM models. While we have only shown beta-binomial inclusion indicator priors for the DTM model, the *MicroBVS* package can support MRF priors for DTM models as well. Additionally, the *MicroBVS* package includes functionality to implement the joint model proposed in [12] and additional code to simulate data for each of these models. Step-by-step worked examples using simulated data are provided in the vignette. Frequentist variable selection methods for microbiome data are covered in Chap. 8.

The computational burden of the models described in this chapter is largely dependent on the dimension of the data, tree complexity, prior specification, and the sparsity of the model. For reference, the DTM model run in the gut microbiome analysis took around 9 h to run 150,000 iterations (0.23 seconds/iteration) on a 2.5 GHz dual-core Intel Core i5 processor with 8 GB RAM. To maintain reasonable computation times and selection performance, the authors in [11] recommend applying DTM models to small-to-medium sized microbiome datasets, that is, with less than 100 compositional components and moderate-to-large tree structures when

$B \times P >> n$. Larger datasets might be analyzed by employing the DM models, which do not incorporate the phylogenetic tree. For comparison, the application of the DM model with beta-binomial priors for inclusion indicators took 24 min (0.14 s/iteration) to run with roughly four times as many taxa (123 versus 28). Using the MRF prior with unknown graphical structure also increases the computation time with larger covariate spaces. For our analysis of the MOMS-PI data, the addition of the Gaussian graphical model increased the computation time to 36 min (0.22 s/iteration). As an avenue for future work, variational inference approaches to DM models have shown promising variable selection results [16].

Acknowledgments We would like to thank Hongzhe Lee for sharing the data from the gut microbiome study with us [31]. Additionally, we would like to thank the owners of the MOMS-PI study data for open-sourcing their data and code for extraction.

References

1. Barbieri, M.M., Berger, J.O., et al.: Optimal predictive model selection. *Ann. Stat.* **32**(3), 870–897 (2004)
2. Brown, P.J., Vannucci, M., Fearn, T.: Multivariate Bayesian variable selection and prediction. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **60**(3), 627–641 (1998)
3. Chen, J., Li, H.: Variable selection for sparse Dirichlet-multinomial regression with an application to microbiome data analysis. *Ann. Appl. Stat.* **7**(1), 418–442 (2013)
4. Dennis III, S.Y.: On the hyper-Dirichlet type 1 and hyper-Liouville distributions. *Commun. Stat.-Theory Methods* **20**(12), 4069–4081 (1991)
5. Eddelbuettel, D., Sanderson, C.: RcppArmadillo: accelerating R with high-performance C++ linear algebra. *Comput. Stat. Data Anal.* **71**, 1054–1063 (2014)
6. Eddelbuettel, D., François, R., Allaire, J., Ushey, K., Kou, Q., Russel, N., Chambers, J., Bates, D.: Rcpp: seamless R and C++ integration. *J. Stat. Softw.* **40**(8), 1–18 (2011)
7. Egozcue, J.J., Pawlowsky-Glahn, V.: Groups of parts and their balances in compositional data analysis. *Math. Geol.* **37**(7), 795–828 (2005)
8. George, E.I., McCulloch, R.E.: Approaches for Bayesian variable selection. *Stat. Sin.* **7**, 339–373 (1997)
9. Integrative, H.: The integrative human microbiome project: dynamic analysis of microbiome-host omics profiles during periods of human health and disease. *Cell Host Microbe* **16**(3), 276 (2014)
10. Knights, D., Parfrey, L.W., Zaneveld, J., Lozupone, C., Knight, R.: Human-associated microbial signatures: examining their predictive value. *Cell Host Microbe* **10**(4), 292–296 (2011)
11. Koslovsky, M.D., Vannucci, M.: MicroBVS: Dirichlet-tree multinomial regression models with Bayesian variable selection - an R package. *BMC Bioinf.* **21**, 301 (2020). <https://doi.org/10.1186/12859-020-03640-0>
12. Koslovsky, M.D., Hoffman, K.L., Daniel, C.R., Vannucci, M.: A Bayesian model of microbiome data for simultaneous identification of covariate associations and prediction of phenotypic outcomes. *Ann. Appl. Stat.* **14**(3), 1471–1492 (2020)
13. La Rosa, P.S., Brooks, J.P., Deych, E., Boone, E.L., Edwards, D.J., Wang, Q., Sodergren, E., Weinstock, G., Shannon, W.D.: Hypothesis testing and power calculations for taxonomic-based human microbiome data. *PLoS One* **7**(12), e52078 (2012)
14. Li, H.: Microbiome, metagenomics, and high-dimensional compositional data analysis. *Ann. Rev. Stat. Appl.* **2**, 73–94 (2015)

15. Li, F., Zhang, N.R.: Bayesian variable selection in structured high-dimensional covariate spaces with applications in genomics. *J. Am. Stat. Assoc.* **105**(491), 1202–1214 (2010)
16. Miao, Y., Kook, J.H., Lu, Y., Guindani, M., Vannucci, M.: Scalable Bayesian variable selection regression models for count data. In: *Flexible Bayesian Regression Modelling*, pp. 187–219. Elsevier, Amsterdam (2020)
17. Minka, T.: The Dirichlet-tree distribution (1999)
18. Newton, M.A., Noueiry, A., Sarkar, D., Ahlquist, P.: Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics* **5**(2), 155–176 (2004)
19. Peterson, C.B., Stingo, F.C., Vannucci, M.: Joint Bayesian variable and graph selection for regression models with network-structured predictors. *Stat. Med.* **35**(7), 1017–1031 (2016)
20. Richardson, S., Bottolo, L., Rosenthal: Bayesian models for sparse regression analysis of high dimensional data. In: *Bayesian Statistics*, vol. 9, pp. 539–569. Oxford University Press, Oxford (2010)
21. Savitsky, T., Vannucci, M., Sha, N.: Variable selection for nonparametric Gaussian process priors: models and computational strategies. *Stat. Sci.: Rev. J. Inst. Math. Stat.* **26**(1), 130–149 (2011)
22. Schwarz, G., et al.: Estimating the dimension of a model. *Ann. Stat.* **6**(2), 461–464 (1978)
23. Shetty, S.A., Hugenholtz, F., Lahti, L., Smidt, H., de Vos, W.M.: Intestinal microbiome landscaping: insight in community assemblage and implications for microbial modulation strategies. *FEMS Microbiol. Rev.* **41**(2), 182–199 (2017)
24. Stingo, F.C., Chen, Y.A., Vannucci, M., Barrier, M., Mirkes, P.E.: A Bayesian graphical modeling approach to microRNA regulatory network inference. *Ann. Appl. Stat.* **4**(4), 2024–2048 (2010)
25. Stingo, F.C., Chen, Y.A., Tadesse, M.G., Vannucci, M.: Incorporating biological information into linear models: a Bayesian approach to the selection of pathways and genes. *Ann. Appl. Stat.* **5**(3), 1978–2002 (2011)
26. Tang, Z.Z., Chen, G., Alekseyenko, A.V., Li, H.: A general framework for association analysis of microbial communities on a taxonomic tree. *Bioinformatics* **33**(9), 1278–1285 (2017)
27. Tang, Y., Ma, L., Nicolae, D.L., et al.: A phylogenetic scan test on a Dirichlet-tree multinomial model for microbiome data. *Ann. Appl. Stat.* **12**(1), 1–26 (2018)
28. Wadsworth, W.D., Argiento, R., Guindani, M., Galloway-Pena, J., Shelburne, S.A., Vannucci, M.: An integrative Bayesian Dirichlet-multinomial regression model for the analysis of taxonomic abundances in microbiome data. *BMC Bioinf.* **18**(1), 94 (2017)
29. Wang, H., et al.: Scaling it up: Stochastic search structure learning in graphical models. *Bayesian Anal.* **10**(2), 351–377 (2015)
30. Wang, T., Zhao, H.: A Dirichlet-tree multinomial regression model for associating dietary nutrients with gut microorganisms. *Biometrics* **73**(3), 792–801 (2017)
31. Wu, G.D., Chen, J., Hoffmann, C., Bittinger, K., Chen, Y.Y., Keilbaugh, S.A., Bewtra, M., Knights, D., Walters, W.A., Knight, R., et al.: Linking long-term dietary patterns with gut microbial enterotypes. *Science* **334**(6052), 105–108 (2011)
32. Xia, Y., Sun, J.: Hypothesis testing and statistical analysis of microbiome. *Genes Dis.* **4**(3), 138–148 (2017)
33. Xu, Z., Knight, R.: Dietary effects on human gut microbiome diversity. *Br. J. Nutr.* **113**(S1), S1–S5 (2015)
34. Zhang, Y., Zhou, H., Zhou, J., Sun, W.: Regression models for multivariate count data. *J. Comput. Graph. Stat.* **26**(1), 1–13 (2017)

A Bayesian Approach to Restoring the Duality Between Principal Components of a Distance Matrix and Operational Taxonomic Units in Microbiome Analyses



Subharup Guha and Somnath Datta

1 Introduction

Recent advances in sequencing technologies along with rapidly declining costs have revolutionized studies of the microbiome. Sequencing the 16S rRNA gene allows scientists to analyze the bacterial compositions of even low-prevalence samples, irrespective of whether they can be easily grown in culture. Using a bioinformatic pipeline such as QIIME [1] or Mothur [8], the large number of sequences is grouped based on similarity into operational taxonomic units (OTUs), with the number of OTUs, p , typically far exceeding the number of samples, n . Since quantitative methods for microbiome data are frequently confronted with the challenges of high dimensionality, there is a critical need for novel analytical techniques for detecting complex structured interactions between the OTUs and samples.

1.1 Motivating Datasets

The authors in [9] have analyzed the bacteria found in $n = 45$ samples obtained from three different kinds of smokeless tobacco products: dry, moist, and brown toombak. Using data from the V4 region of the 16S rRNA gene consisting of more than 3 million observed sequences, the QIIME pipeline was utilized to group the sequences into 5345 OTUs. A thresholding criterion was then applied to select $p = 271$ OTUs while retaining nearly 95% of the observed sequences. The main challenge is to find a lower dimensional representation of the data capable of

S. Guha (✉) · S. Datta
Department of Biostatistics, University of Florida, Gainesville, FL, USA
e-mail: s.guha@ufl.edu; somnath.datta@ufl.edu

successfully differentiating the tobacco types, while also determining which OTUs are the most influential.

The MetaSUB International Consortium and the Scientific Programme Committee of the conference on Critical Assessment of Massive Data Analysis (CAMDA 2018) collaborated on a study on forensic metagenomics involving multiple cities. The subway dataset covers several international cities, with tens of samples per city. After quality control, we focused on three cities with the largest number of samples. This yielded a dataset consisting of 60, 34, and 30 samples from Porto Santo Island (Portugal), Sacramento, and New York City, respectively. For these $n = 124$ samples, the aggregated species counts for $p = 898$ OTUs were normalized to give a matrix of relative abundances. A key challenge question with this dataset was that of identification of the city source of a microbiome sample, even though a complete analysis was also desired.

In general, the raw OTU counts representing the abundances of a taxonomic rank can be arranged in a matrix whose n rows correspond to observations or sample units and whose p columns correspond to species or OTUs, with $n \ll p$. In microbiome analysis, it is a common practice to transform the matrix of OTU counts in a variety of different ways, such as (1) the raw OTU counts are scaled so that the row sums are equal, say, 100; (2) the rows are centered so that their averages equal 0; and (3) the columns are centered so that the column averages are all equal to 0. Applying one or more of such transformations gives an $n \times p$ *modified abundance matrix* denoted by X .

1.2 Nonlinear or Stochastic Distances

In several ecological studies, the species abundance data are also used to calculate an $n \times n$ dissimilarity or distance matrix Δ whose (i, k) th element represents the dissimilarity between the i th and k th observations. We refer to these dissimilarities as “distances” without insisting that they satisfy the triangle inequality. Matrix Δ may be a highly nonlinear function of the species abundance data and may rely on complementary information such as phylogeny of the microbial communities. The relationship could be even more complicated, as in UniFrac distances, which are stochastic functions of the abundance data and phylogenetic trees.

The distance measures commonly used in ecology (e.g., see [5]) offer effective descriptions of the observations. Specifically, the first few principal components (PCs) of matrix Δ partition the observations into interpretable groups across which the OTUs systematically vary. These partitions often represent interpretable groups. For example, the authors in [9] found that UniFrac distances are quite successful at differentiating the three tobacco types in the motivating dataset, while also allowing replicates of the same product to be closely clustered. Gower distances are also able to discriminate between clusters [4]. However, it is difficult to identify the species or OTUs that significantly contributed to the distances between the observations. We cannot make a biplot that uses the UniFrac or Gower PCs to indicate which OTUs are the most successful in predicting tobacco type.

1.3 Limitations of SVD-Based Approaches

Important linear combinations of the features or OTUs are represented by the PCs of $X^T X$ representing the covariance matrix of the OTUs for the modified abundance matrix. As is well known, these PCs are also available from a singular value decomposition (SVD) of X . The latter procedure yields a set of singular vectors for the observations and a different set of singular vectors for OTUs. This “duality” of the two sets of singular vectors in SVD has useful advantages. For example, the factor loadings, i.e., coefficients of the singular vectors in feature space, are available for each component in observation space and may be used to ascertain which OTUs have the largest contributions. The singular vectors in observation space may be used as model predictors because they are available for future observations. However, since the latter set of singular vectors are the eigenvectors of $X X^T$, a potential disadvantage is that we are implicitly using $X X^T$ as a measure of similarity between the observations even though alternative distance measures such as Gower may be more appropriate in microbiome analyses.

The goal of this chapter is to develop a Bayesian approach that restores, through an approximate SVD-type decomposition of the modified abundance matrix X , the duality between the set of singular vectors of an arbitrary distance matrix Δ and a set of singular vectors in feature space. The authors in [7] have proposed several optimization-based approximate decompositions that link the PCs of Δ with linear combinations of OTU frequencies. However, although the numerical algorithms recommended by [7] are found to rapidly converge to a solution, there is no guarantee that the solution represents the global rather than local minimum of the high-dimensional objective function. This is problematic because high-dimensional objective functions typically possess large numbers of modes. Furthermore, uncertainty estimates are not available by these approaches.

Motivated by this, we have adopted a Bayesian approach where the assumption of a squared error loss function results in the posterior mean being the optimal estimate of the unknown parameters. This offers a key advantage because compared to optimization-based approaches that may get stuck in local modes, a fast-mixing MCMC chain provides relatively robust inferences about the posterior means of the parameters. This approach also readily provides uncertainty estimates for the model parameters and their functionals, such as standard errors and credible intervals.

The rest of the chapter is organized as follows. A Bayesian model for the modified abundance matrix X is constructed in Sect. 2. In Sect. 3, we discuss how the special structure of the model can be exploited to partition the model sum of squares, construct a scree plot depicting the individual OTU contributions, and make biplots for arbitrary distances. Since the posterior distribution is analytically intractable for estimation purposes, Sect. 4.1 develops a Markov chain Monte Carlo (MCMC) technique for generating posterior samples of the model parameters.

Although accurate, the fully Bayesian method can be computationally intensive for even moderately large microbiome datasets. Section 4.2 develops an efficient version called the Skinny Bayesian technique which, although approximate, is

designed to mimic the inferences of the fully Bayesian method at a fraction of the computational cost. This makes the Skinny Bayesian technique an attractive option for microbiome analyses. Section 5 demonstrates the effectiveness of the proposed methodologies by a simulation study. In Sect. 6, we analyze the two microbiome datasets, where the proposed Bayesian methodology has the important advantage of not only providing point estimates for data with highly multimodal posterior densities but also facilitating interval and posterior density estimation of all the model parameters. We find that alternative distance matrices, such as Gower and UniFrac, that the technique facilitates have lower classification test error rates than the covariances assumed by default by traditional PC analyses.

2 A Bayesian Formulation

The data consist of not only the transformed OTU frequencies of modified abundance matrix X but also distance matrix Δ ; that is, $\mathcal{D} = (X, \Delta)$ represents the outcomes on which posterior inferences will be based. Let the rank of distance matrix Δ be q , so that $q \leq n$; the rank of X could, of course, be different. Since matrix Δ is symmetric and real, it may be expressed as $\Delta = \mathbf{B}\mathbf{E}\mathbf{B}^T$, where \mathbf{B} is an $n \times q$ matrix with orthonormal columns and \mathbf{E} is a $q \times q$ diagonal matrix with nonzero diagonal elements.

The relation between modified abundance matrix X and distance matrix Δ is driven by the specifics of the application and may be complicated. At one extreme, when the distance matrix is the correlation matrix, as in SVD-based approaches, Δ is completely determined by X . At the other extreme, such as ecological investigations involving UniFrac distances, Δ is a complicated function of X , external inputs, and independent stochastic variables.

All Bayesian inferences on model parameters and their functionals are based on the posterior. However, unlike most Bayesian applications, we will not specify a hierarchical model and so arrive at the posterior distribution as the normalized product of a likelihood and prior. This is mainly because, as previously noted, the specification of a likelihood function for the data \mathcal{D} is typically not straightforward in microbiome analyses involving alternative distances. Instead, our approach is loosely related to recent work by [6] that replaces the log-likelihood in Bayesian probabilistic clustering by a loss function.

As a first step, if q^* denotes the rank of X , we note that the SVD decomposition of abundance matrix X is $\mathbf{A}\mathbf{G}\mathbf{C}^T$, where \mathbf{A} is an $n \times q^*$ matrix with orthonormal columns, \mathbf{G} is a $q^* \times q^*$ diagonal matrix with positive diagonal entries, and \mathbf{C} is a $q^* \times p$ matrix with orthonormal columns. The Frobenius norm of $X - \mathbf{A}\mathbf{G}\mathbf{C}^T$ is thus zero.

To accommodate the general situation where the distance Δ is not the covariance or a deterministic function of X , we utilize information extracted from matrix Δ to devise an approximate SVD-type representation of X . Specifically, using the known eigenmatrix \mathbf{B} of distance matrix Δ , we construct a posterior whose mode matches

the global minimum of the Frobenius norm:

$$\mathcal{F}(\mathbf{V}, \mathbf{D}) = \|\mathbf{X} - \mathbf{B}\mathbf{D}\mathbf{V}^T\|_F^2, \quad (1)$$

for an arbitrary $p \times q$ orthonormal matrix \mathbf{V} and $q \times q$ diagonal matrix \mathbf{D} with arbitrary nonnegative elements. Information about distance matrix Δ is thus incorporated into the posterior via the eigenmatrix \mathbf{B} . The motivation for the global minimum requirement is that, unlike the SVD decomposition, an exact zero of objective function (1) is not generally possible for any choice of orthonormal matrix \mathbf{V} and diagonal matrix \mathbf{D} . As described in detail in Sect. 3, the approximate decomposition of modified abundance matrix \mathbf{X} given by (1) may then be used to select important OTUs and partition the variability in \mathbf{X} into contributions corresponding to the PCs of an arbitrary distance matrix Δ between observations. Section 4.3 describes how the model parameters are estimated by their posterior means which, in turn, are estimated using an MCMC sample.

2.1 Posterior Density

Given a “precision” parameter $\tau^2 > 0$, consider the function

$$g(\mathbf{V}, \mathbf{D}, \tau^2) = \tau^{np} \exp(-\tau^2 \mathcal{F}(\mathbf{V}, \mathbf{D})/2), \quad (2)$$

for $\mathbf{V} \in \mathcal{S}_{q,p}$, $\mathbf{D} \in \mathcal{D}_q^+$, and $\tau^2 > 0$,

where $\mathcal{S}_{q,p}$ represents the Stiefel manifold, that is, the set of $p \times q$ matrices with orthonormal columns, and \mathcal{D}_q^+ is the set of $q \times q$ diagonal matrices with nonnegative elements. The posterior density is defined as the normalized version of the aforementioned function:

$$\pi(\mathbf{V}, \mathbf{D}, \tau^2 | \mathfrak{D}) = c(\mathfrak{D}) \cdot g(\mathbf{V}, \mathbf{D}, \tau^2), \quad (3)$$

for $\mathbf{V} \in \mathcal{S}_{q,p}$, $\mathbf{D} \in \mathcal{D}_q^+$, and $\tau^2 > 0$,

where $c(\mathfrak{D})$ is the normalizing constant.

Now, consider the conditional posterior

$$\pi(\mathbf{V}, \mathbf{D} | \tau^2, \mathfrak{D}) = g(\mathbf{V}, \mathbf{D}, \tau^2) / \int_{\mathcal{D}_q^+} \int_{\mathcal{S}_{q,p}} g(\mathbf{V}, \mathbf{D}, \tau^2) d\mathbf{V} d\mathbf{D}.$$

For every fixed value of τ^2 , it can be easily verified that the mode of conditional posterior $\pi(\mathbf{V}, \mathbf{D} | \tau^2, \mathfrak{D})$ is the minimizer of objective function $\mathcal{F}(\mathbf{V}, \mathbf{D})$. This implies that the minimizer of $\mathcal{F}(\mathbf{V}, \mathbf{D})$ coincides with the (\mathbf{V}, \mathbf{D}) component of the full posterior’s mode in equation (3).

3 Model Sum of Squares and Biplots

Due to the form of posterior density (3), whose mode minimizes Frobenius norm (1), we define as the *model sum of squares* the part of the modified abundance matrix X that can be explained by the model parameters:

$$\mathcal{M}(\mathbf{V}, \mathbf{D}) = \|\mathbf{B}\mathbf{D}\mathbf{V}^T\|_F^2.$$

Furthermore, we define

$$\text{Percent variation in } X \text{ explained by the model} = \{\mathcal{M}(\mathbf{V}, \mathbf{D}) / \|X\|_F^2\} \times 100\%. \quad (4)$$

An estimate of the marginal posterior density of $\mathcal{M}(\mathbf{V}, \mathbf{D})$ is available from the Bayesian analysis. Under squared error loss, the marginal posterior mean provides an optimal estimate, $\hat{\mathcal{M}}(\mathbf{V}, \mathbf{D})$, of the model sum of squares, and the posterior standard deviation provides an uncertainty estimate.

Since matrices \mathbf{V} and \mathbf{B} are orthonormal, the model sum of squares can be conveniently partitioned for the purposes of constructing scree plots and biplots. For example, it can be decomposed into the parts explained by each OTU:

$$\begin{aligned} \mathcal{M}(\mathbf{V}, \mathbf{D}) &= \sum_{j=1}^p w_j^2, \quad \text{where} \\ w_j^2 &= \sum_{t=1}^q v_{jt}^2 d_t^2, \end{aligned} \quad (5)$$

so that data from the j th OTU contributes w_j^2 to the model sum of squares.

Alternatively, the model sum of squares may be partitioned into parts explained by the q PCs of distance matrix Δ :

$$\mathcal{M}(\mathbf{V}, \mathbf{D}) = \sum_{t=1}^q d_t^2. \quad (6)$$

Since they are functions of the model parameters, these partitions can be estimated along with their posterior credible intervals.

A two-dimensional biplot for distance matrix Δ can be plotted as follows. From equation (6), we observe that the first two PCs of distance matrix Δ are the columns of orthonormal matrix \mathbf{V} that correspond to the largest and second largest diagonal elements of matrix \mathbf{D} . For each PC, the scores of the n observations are obtained by projecting the n rows of matrix X onto the direction represented by the PC. Figure 5 displays the biplot for the subway dataset with the scores plotted in blue; the symbols represent the three cities from which the samples were drawn.

Additionally, to display the effects of the top m OTUs, we first apply Eq. (5) to identify the m OTUs with the greatest contributions and then plot the values of the matching elements of the two PCs along perpendicular axes of the biplot. Figure 5 displays the top $m = 5$ OTUs for the subway dataset.

4 Posterior Inference

Since the posterior is analytically intractable, we explore the posterior distribution using MCMC techniques.

4.1 Gibbs Sampler

It is possible to iteratively sample from the full conditional distributions of the model parameters, i.e., matrices \mathbf{V} and \mathbf{D} , and scalar τ^2 , resulting in a Gibbs sampler whose post-burn-in draws are distributed as posterior (3). The Gibbs sampler relies on this simple but important result whose proof is given in the Appendix.

Lemma 1 *Let \mathbf{B}_0 be an $n \times (n - q)$ matrix with orthonormal columns belonging to the null space of the columns of matrix \mathbf{B} . Objective function $\mathcal{F}(\mathbf{V}, \mathbf{D})$ defined in Eq. (1) has the equivalent expression*

$$\mathcal{F}(\mathbf{V}, \mathbf{D}) = \sum_{j=1}^q d_j^2 - 2\text{tr}(\mathbf{D} \mathbf{Q}^T \mathbf{V}) + \text{tr}(\mathbf{Q}^T \mathbf{Q}) + \|\mathbf{B}_0^T \mathbf{X}\|_F^2, \quad (7)$$

where the $p \times q$ matrix, $\mathbf{Q} = \mathbf{X}^T \mathbf{B}$.

Applying Lemma 1, the model parameters are iteratively generated as follows:

1. Full conditional of matrix \mathbf{V} :

$$[\mathbf{V} | \mathbf{D}, \tau^2, \mathcal{D}] \propto \exp \text{tr}(\tau^2 \mathbf{D} \mathbf{Q}^T \mathbf{V}),$$

which is the von Mises–Fisher matrix distribution [2] with parameter $\tau^2 \mathbf{Q} \mathbf{D}$. Refer to [3] for a fast-mixing Gibbs sampler when $p > q$, as in this application. The Hoff algorithm relies on the availability of a Monte Carlo sampler for vector von Mises–Fisher distributions, e.g., via command `rvmf` implemented in the `Directional` R package.

2. Full conditional of matrix \mathbf{D} :

Let $\mathbf{R} = \mathbf{Q}^T \mathbf{V}$ with diagonal elements $r_{11}, r_{22}, \dots, r_{qq}$. Then, the diagonal elements of matrix \mathbf{D} are independent with truncated normal distributions:

$$d_j \mid \mathbf{V}, \tau^2, \mathfrak{D} \stackrel{\text{indep}}{\sim} N(r_{jj}, \tau^{-2}) \cdot \mathcal{I}(d_j \geq 0), \quad j = 1, \dots, q. \quad (8)$$

3. Full conditional of precision parameter τ^2 : We obtain

$$\tau^2 \mid \mathbf{D}, \mathbf{V}, \mathfrak{D} \sim \text{gamma}(np/2 + 1, \mathcal{F}(\mathbf{V}, \mathbf{D})/2), \quad (9)$$

which is parameterized with mean equal to $(np + 2)/\mathcal{F}(\mathbf{V}, \mathbf{D})$.

4.2 Dimension Reduction: Skinny Bayesian Technique

Posterior inferences by the aforementioned method can be computationally intensive. The computationally intensive step is the Gibbs sampler for the von Mises–Fisher matrix full conditional of matrix \mathbf{V} ; the costs of generating the remaining parameters are trivial. For example, in the motivating Tobacco dataset, we have $p = 271$ OTUs and $n = 45$ samples, and the rank of dissimilarity matrix Δ is $q = 45$. Even this moderate-sized dataset imposes a heavy computational burden on the Gibbs sampler.

We wish to develop a computationally efficient methodology that, even if it is not exact, closely matches the inferences of posterior density (3). Suppose that the columns of modified abundance matrix \mathbf{X} are partitioned into two submatrices, n by p_1 matrix \mathbf{X}_1 and n by $(p - p_1)$ matrix \mathbf{X}_2 , where $n < p_1 < p$. The key idea is to select a submatrix \mathbf{X}_1 having far fewer columns than matrix \mathbf{X} (i.e., $p_1 \ll p$), but which still captures most of the column variability of matrix \mathbf{X} . The reduced dimension matrix, \mathbf{X}_1 , could then be used for approximate posterior inferences at a fraction of the computational cost. We refer to this technique as the *Skinny Bayesian* technique to distinguish it from the fully Bayesian technique of Sect. 4.1. The details are given below.

4.2.1 Subsetted Data Matrix

In order to select a subset of the columns of modified abundance matrix \mathbf{X} that captures most of the variability, we want

$$\|\mathbf{X}_1\|_F^2 / \|\mathbf{X}\|_F^2 = r_0^2, \quad (10)$$

with r_0^2 chosen as close as possible to 1 given the computing resources. To achieve this aim, we first rank the individual columns with respect to their Frobenius norm:

$$\|\mathbf{x}_{(1)}\|_F^2 \geq \|\mathbf{x}_{(2)}\|_F^2 \dots \geq \|\mathbf{x}_{(p)}\|_F^2$$

and evaluate

$$p_1 = \max \left[n + 1, \quad \operatorname{argmin}_{j=1,\dots,p} \left\{ j : \sum_{t=1}^j \left(\|\mathbf{x}_{(t)}\|_F^2 / \|X\|_F^2 \right) \geq r_0^2 \right\} \right], \quad (11)$$

so that $q \leq n < p_1$. The reduced data matrix is then defined as

$$\mathbf{X}_1 = [\mathbf{x}_{(1)}, \dots, \mathbf{x}_{(p_1)}].$$

Typically, $p_1 \ll p$. The remaining $(p - p_1)$ columns of matrix \mathbf{X} are collectively denoted by \mathbf{X}_2 .

4.2.2 Lower Dimensional Parameters and Induced Posterior

We match the dimensions of the reduced data to define the matrix \mathbf{V}_1 be a $p_1 \times q$ orthonormal matrix with fewer rows than $p \times q$ orthonormal matrix \mathbf{V} ; that is, \mathbf{V}_1 takes values in the lower dimensional Stiefel manifold \mathcal{S}_{q,p_1} . The diagonal matrix \mathbf{D} with nonnegative elements is as defined before. We assume the following relationship between the two orthonormal matrices:

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_1 \\ \mathbf{0} \end{bmatrix}, \quad (12)$$

where $\mathbf{0}$ denotes a matrix of $p - p_1$ rows and q columns with all zero entries. Since matrix \mathbf{V}_1 consists of q orthonormal columns, this guarantees that matrix \mathbf{V} is also orthonormal.

Instead of minimizing objective function (1), the estimation of matrices \mathbf{V}_1 and \mathbf{D} is motivated by minimizing the following objective function involving the reduced data:

$$f_d^{(1)}(\mathbf{V}_1, \mathbf{D}) = \|\mathbf{X}_1 - \mathbf{B}\mathbf{D}\mathbf{V}_1^T\|_F^2. \quad (13)$$

An analogous result to Lemma 1 immediately follows

$$f_d^{(1)}(\mathbf{V}_1, \mathbf{D}) = \sum_{j=1}^q d_j^2 - 2\operatorname{tr}(\mathbf{D}\mathbf{Q}_1^T\mathbf{V}_1) + \operatorname{tr}(\mathbf{Q}_1^T\mathbf{Q}_1) + \|\mathbf{B}_0^T\mathbf{X}_1\|_F^2, \quad (14)$$

where the $p_1 \times q$ matrix, $\mathbf{Q}_1 = \mathbf{X}_1^T\mathbf{B}$. For any parameters satisfying Eq. (12), we then obtain

$$\mathcal{F}(\mathbf{V}, \mathbf{D}) = f_d^{(1)}(\mathbf{V}_1, \mathbf{D}) + \|\mathbf{X}_2\|_F^2, \quad (15)$$

so that

$$\{\mathcal{F}(\mathbf{V}, \mathbf{D}) - f_d^{(1)}(\mathbf{V}_1, \mathbf{D})\}/\|\mathbf{X}\|_F^2 = 1 - r_0^2.$$

The special structure imposed by equation (12) implies that the corresponding minimum of function $\mathcal{F}(\mathbf{V}, \mathbf{D})$ is smaller when matrix \mathbf{V} is allowed to assume unrestricted values on the higher dimensional Stiefel manifold $\mathcal{S}_{q,p}$, as in Sect. 2. However, since by design matrix \mathbf{X}_1 that accounts for most of the variation in matrix \mathbf{X} , we expect the difference between the two minima to be small.

The posterior induced by restriction (12) on the set of lower dimensional parameters is

$$\begin{aligned} \pi_1(\mathbf{V}_1, \mathbf{D}, \tau^2 | \mathcal{D}) &\propto g_1(\mathbf{V}_1, \mathbf{D}, \tau^2), \quad \text{where} \\ g_1(\mathbf{V}_1, \mathbf{D}, \tau^2) &= \tau^{np} \exp(-\tau^2 \mathcal{F}(\mathbf{V}, \mathbf{D})/2), \\ &= \tau^{np} \exp(-\tau^2 f_d^{(1)}(\mathbf{V}_1, \mathbf{D})/2) \exp(-\tau^2 \|\mathbf{X}_2\|_F^2/2), \quad \mathbf{V}_1 \in \mathcal{S}_{q,p_1}, \end{aligned} \quad (16)$$

by Eq. (15).

4.2.3 Faster Inference Procedure

Since $q \leq n < p_1$, the induced lower dimensional posterior (16) has a similar structure as unrestricted posterior (3), and it is straightforward to iteratively sample from the full conditional distributions of the parameters $(\mathbf{V}_1, \mathbf{D}, \tau^2)$. The important advantage associated with analyzing restricted posterior (16) is the drastically reduced computational cost.

1. **Full conditional of matrix \mathbf{V}_1 :** Upon applying Eq. (14), posterior (16) gives

$$\begin{aligned} [\mathbf{V}_1 | \mathbf{D}, \tau^2, \mathcal{D}] &\propto \exp(-\tau^2 f_d^{(1)}(\mathbf{V}_1, \mathbf{D})/2) \\ &\propto \exp \text{tr}(\tau^2 \mathbf{D} \mathbf{Q}_1^T \mathbf{V}_1), \end{aligned}$$

which is the von Mises–Fisher matrix distribution with parameter $\tau^2 \mathbf{Q}_1 \mathbf{D}$.

Since $p_1 > q$, the Gibbs sampler of [3], previously mentioned in Sect. 4.1, could be applied to generate MCMC draws. However, unlike Sect. 2, matrix \mathbf{V}_1 consists of only p_1 rows instead of p ($\gg p_1$) rows for matrix \mathbf{V} . This results in drastically reduced computational costs.

2. **Full conditional of matrix \mathbf{D} :** Analogously to Sect. 4.1, relation (12) gives $\mathbf{R} = \mathbf{Q}^T \mathbf{V} = \mathbf{Q}_1^T \mathbf{V}_1$. The diagonal elements of matrix \mathbf{D} are then updated as specified in expression (8).
3. **Full conditional of parameter τ^2 :** This is updated as in expression (9) with the quantity $\mathcal{F}(\mathbf{V}, \mathbf{D})$ computed using Eq. (15).

4.3 Model Parameter Estimates

The MCMC sample is post-processed to compute empirical average estimates of the posterior means of the model parameters. The exception is orthonormal matrix V , which is defined on Stiefel manifold $\mathcal{S}_{q,p}$. There is no guarantee that the posterior mean, $E[V|\mathcal{D}]$, belongs to the same space. Consequently, we estimate V as $\operatorname{argmin}_{V \in \mathcal{S}_{q,p}} \|V - E[V|\mathcal{D}]\|_F^2$ representing the point on the Stiefel manifold closest in Frobenius norm to the posterior mean. To obtain this estimate, the MCMC sample first gives an empirical average estimate of $E[V|\mathcal{D}]$, denoted by \tilde{V} . The MCMC sample is then processed a second time to compute $\operatorname{argmin}_{V \in \mathcal{S}_{q,p}} \|V - \tilde{V}\|_F^2$.

5 Simulation Study

To evaluate the fully Bayesian technique's ability to accurately infer the underlying model parameters, we analyzed 50 artificial datasets.

5.1 Generation Strategy

For $n = 45$ samples and $p = 150$ OTUs, we independently generated 50 datasets by the following sequence of steps:

1. **Rank q of distance matrix Δ :** Set the rank q equal to either 38 or 39 with probability 0.5.
2. **Matrix B :** Generate q orthonormal vectors of length n , denoted by b_1, \dots, b_q . Let the $n \times q$ matrix, $B = [b_1, \dots, b_q]$.
3. **Diagonal elements of true matrix D_0 :** For $j = 1, \dots, q$, generate $d_{j0} = |Z_j|$, where $Z_j \stackrel{iid}{\sim} N(0, 10^2)$. Then, $D_0 = \operatorname{diag}\{d_{10}, \dots, d_{q0}\}$.
4. **True orthonormal matrix V_0 :** Generate the $n \times q$ matrix, $U = ((U_{ij}))$, where the random variables U_{ij} are iid standard normal. Orthonormal matrix V_0 is obtained by the Gram–Schmidt orthogonalization of the q columns of matrix U .
5. **Data matrix X :** Let $X = BD_0V_0^T + \Upsilon$, where Υ is an $n \times p$ matrix of iid normal errors with mean 0 and variance τ_0^{-2} , where true precision $\tau_0 = 5$.

Notice that the data generation strategy relies on distance matrix Δ only through its rank q and matrix of eigenvalues B ; it is not necessary to generate Δ itself. For each dataset, we implemented the inferential technique described in Sect. 2 to generate a post-burn-in MCMC sample. The parameters were estimated by empirical average estimates computed from the MCMC sample.

R code implementing the simulation procedure is available in the folder “BDVt.zip” available at <https://github.com/sguha-lab/Microbiome-SVD>. To generate the data, analyze the data using the aforementioned MCMC procedure,

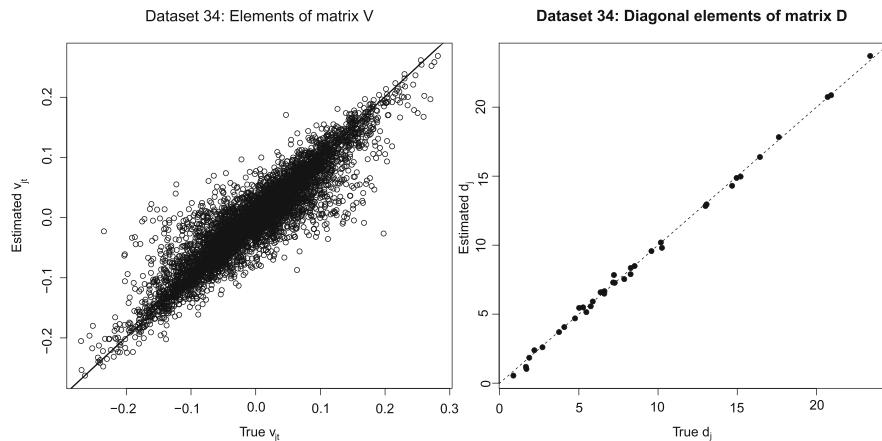


Fig. 1 For simulated dataset 34, plots of actual versus estimated model parameters and observations. A 45° line has been added for reference

Table 1 For the 50 artificial datasets, five-number summary of the correlations between the true and estimated matrix elements

	Minimum	First quantile	Median	Third quantile	Maximum
$\mathbf{V} = ((v_{jt}))$	0.803	0.847	0.872	0.890	0.918
$\mathbf{D} = \text{diag}((d_j))$	0.998	0.999	0.999	0.999	1.000

and obtain the posterior inferences described herein, simply extract the files from the zip folder, open the project “BDVt.Rproj” from RStudio, and type

```
source ("main.R")
```

in the RStudio console window.

For a randomly selected simulated dataset, namely dataset 34, Fig. 1 displays the actual versus estimated elements of parameter matrices \mathbf{V} and \mathbf{D} . The plots reveal the high degree of inferential accuracy for dataset 34. More generally, summarizing over the 50 datasets, the correlations between the true and estimated matrix elements are presented in Table 1. The correlations were found to be high in all the datasets, although they were relatively low for the high-dimensional matrix \mathbf{V} consisting of $pq \geq 150(38) = 5,700$ elements; unlike some other model parameters, the estimation of \mathbf{V} is not guaranteed to be consistent as the number of samples and the number of OTUs grow. Consequently, posterior inferences of the matrix \mathbf{V} elements may be imprecise even for large datasets.

The left panel of Fig. 2 plots the histogram of estimates, $\hat{\tau}$, which are equal to the estimated posterior means in each of the 50 datasets. The marginal posterior density of τ for dataset 34 has been plotted in the right panel of Fig. 2, with the shaded region representing the equal-tailed 95% Bayesian credible interval. For all 50 datasets, histograms of the lower and upper limits of the 95% Bayesian credible intervals for τ are displayed in Fig. 3. Obviously, an indication of inferential accuracy is that a

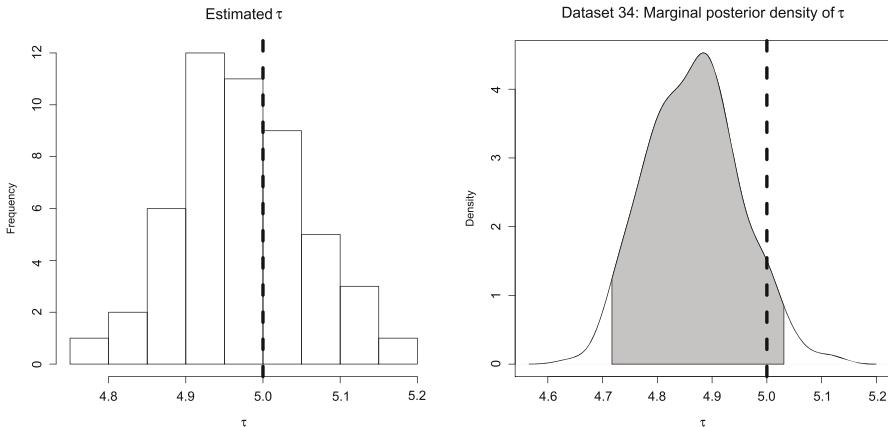


Fig. 2 Posterior inference of τ . Histogram of $\hat{\tau}$ for the 50 datasets in the simulation study is displayed on the left. The estimated marginal posterior density of τ in simulated dataset 34 is displayed on the right. The dashed vertical line represents the common true value of $\tau_0 = 5$ in both plots. The shaded region in the right panel is a 95% Bayesian credible interval with posterior probability of 0.025 in each tail

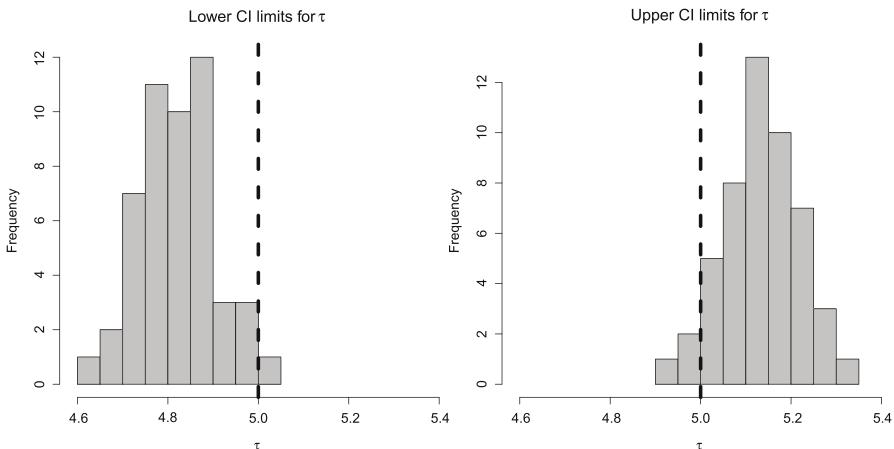


Fig. 3 For the 50 artificial datasets, histograms of the lower and upper limits of 95% Bayesian credible intervals are displayed in the left and right panels, respectively. The dashed vertical lines represent the common true value of $\tau_0 = 5$

large proportion of the 50 lower (upper) limits is less (greater) than the true value of $\tau_0 = 5$, which is shown by the dashed vertical line in each plot. Summarizing over the 50 datasets, the percentage of 95% credible intervals containing the true value of $\tau_0 = 5$ was 46/50 or 92%.

For dataset 34, Table 2 displays the true values of the diagonal matrix elements, d_1, \dots, d_{38} , along with the lower and upper limits of their inferred 95% Bayesian

Table 2 For simulated dataset 34, a comparison of the true values and 95% Bayesian credible intervals for the nonzero elements of the 38×38 diagonal matrix, \mathbf{D} . The true parameter values are displayed in the middle column of numbers. The end points of the credible intervals are given in the first and third columns. Marked in bold are the four parameters whose true values lie outside the credible intervals

Parameter	2.5th percentile	Truth	97.5th percentile
d_1	23.30	23.39	24.12
d_2	20.48	20.93	21.34
d_3	20.32	20.71	21.12
d_4	17.46	17.62	18.20
d_5	16.01	16.43	16.74
d_6	14.61	15.21	15.40
d_7	14.55	14.96	15.28
d_8	13.92	14.68	14.71
d_9	12.52	13.06	13.44
d_{10}	12.41	13.00	13.30
d_{11}	9.43	10.25	10.24
d_{12}	9.81	10.18	10.59
d_{13}	9.14	9.58	9.97
d_{14}	8.11	8.52	8.85
d_{15}	7.95	8.28	8.69
d_{16}	7.51	8.27	8.26
d_{17}	7.16	7.87	7.95
d_{18}	6.87	7.27	7.74
d_{19}	7.42	7.22	8.22
d_{20}	6.87	7.17	7.66
d_{21}	6.28	6.62	7.06
d_{22}	6.12	6.61	6.82
d_{23}	6.16	6.52	6.97
d_{24}	6.18	6.37	7.00
d_{25}	5.54	5.88	6.30
d_{26}	5.18	5.75	6.00
d_{27}	4.78	5.47	5.56
d_{28}	5.11	5.27	5.94
d_{29}	5.08	5.02	5.87
d_{30}	4.32	4.76	5.09
d_{31}	3.63	4.08	4.49
d_{32}	3.33	3.75	4.15
d_{33}	2.15	2.70	3.11
d_{34}	1.90	2.19	2.91
d_{35}	1.31	1.87	2.37
d_{36}	0.08	1.69	1.91
d_{37}	0.40	1.66	1.82
d_{38}	0.03	0.87	1.02

credible intervals. Marked in bold are the 4 elements for which the true values did not belong to the credible intervals. This corresponds to an interval estimation accuracy of $1 - 4/38$, or 89.5%, for dataset 34. For the 50 datasets, the average interval estimation accuracy was 93.9% and the standard deviation was 4.2%. For the $pq = 150(38) = 5,700$ elements of matrix V , the interval estimation accuracy was 95.4% for dataset 34. For the 50 datasets, the interval estimation accuracy for the matrix V elements averaged 95.0% with a standard deviation of 0.4%.

These results demonstrate the reliability of the fully Bayesian procedure for the artificially generated datasets.

6 Data Analysis

We return to the two microbiome datasets to analyze them using the proposed Bayesian methodology.

6.1 Tobacco Data

We analyzed the Tobacco dataset of [9] using the fully Bayesian and Skinny Bayesian techniques described, respectively, in Sects. 4.1 and 4.2. The rows of the abundance matrix were first converted to percent abundances to eliminate differences in scale and then centered the rows and columns to sum to zero. To study the effects of each OTU in a biplot, the matrix columns were standardized to unit variance. With $r_0^2 = 0.99$ in Eq. (10), we obtained a reduced data matrix X_1 by the Skinny Bayesian method. The reduced data matrix accounted for 99% of the column variability in data matrix X and consisted of $p_1 = 116$ OTUs in Eq. (11). This resulted in a 86.9% *reduction* in the computational costs relative to the fully Bayesian method.

To compare the ability of the two Bayesian methods to explain the modified abundance matrix X , we evaluated the percent variation in X , defined in (4). They were estimated to be 90.6% and 90.1%, respectively, for the Fully Bayesian and Skinny Bayesian methods. The methods display excellent, and almost identical, performance with respect to both sets of measures. Additionally, the posterior standard deviations of the model sums of squares provide uncertainty estimates for the estimated percent variation, which were 0.06% and 0.1%, respectively.

Applying Eq. (5), scree plots for the model variance explained by each OTU are displayed in Fig. 4. For each method, the relative contributions of the top 20 OTUs with the largest contributions have been plotted along with 95% posterior credible intervals. The almost negligible differences between the estimates provided by the two methods demonstrate the effectiveness of the Skinny Bayesian technique. Specifically, the top OTUs identified by the two Bayes methods are identical. Table 4 displays the Greengenes identification number, family, genus, and species

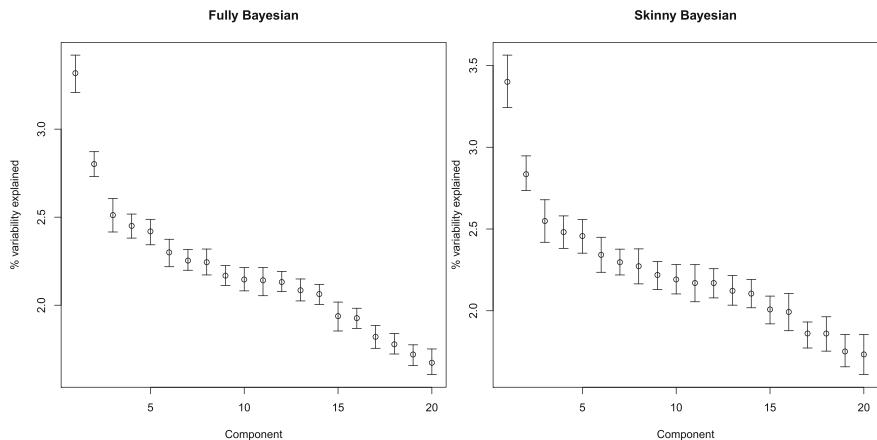


Fig. 4 Scree plots for variance explained by each OTU. For the Fully Bayesian approach (left panel) and Skinny Bayesian approach (right panel), the contributions of the top 20 (out of 271) OTUs having the largest contributions are plotted along with 95% posterior credible intervals for the OTU contributions

of the top 10 of these OTUs. The top two OTUs, 2336242 and 4397708, have significantly greater contributions than the remaining OTUs. Since they belong to the families *Bacillaceae* and *Lactobacillaceae*, respectively, they are biologically distant. However, they are more closely related to several other OTUs in Fig. 4 and Table 4.

A comparison of the standard errors for the percent variation in X and credible interval widths in Fig. 4 reveals the greater uncertainty (i.e., lower precision) associated with the Skinny Bayesian method. Although this is a common feature of many dimension reduction procedures, which are associated with loss of information, the precisions in this case are only slightly lower than those of the fully Bayesian method and are more than compensated by the substantial reductions in the computational costs.

Next, we compared the success of the first two PCs of covariance, UniFrac, and Gower distances in classifying the three types of smokeless tobacco products (dry, moist, and brown toombak). For distances different from the covariance matrix, the procedure for obtaining the PCs is described in Sect. 3. The $n = 45$ samples were split into training and test cases in a 9:1 ratio. A K-nearest neighbor classifier, with the optimal K chosen by cross-validation, was utilized to train the classifier and predict the test sample labels. The error rate was estimated on the basis of 500 independent random splits. The test error rates along with uncertainty estimates are presented in Table 3. We find that the greatest accuracy (i.e., lowest test error rate) is achieved by Gower distances followed by UniFrac distances. The results demonstrate the practical advantages of using alternative distance matrices (Table 4).

Table 3 For the Tobacco dataset, test error rates for classifying the types of smokeless tobacco products using the first two PCs under different distance matrices. See the text in Sect. 6 for further explanation.

Distance Δ	Test error rate	
	Estimate	Standard error
Covariance	35.76%	0.86%
UniFrac	31.76%	0.75%
Gower	29.84%	0.78%

Table 4 Greengenes identification number, family, genus, and species of the top 10 OTUs with the largest contributions identified in Fig. 4

OTU	Family	Genus	Species
2336242	Bacillaceae	Unknown	Unknown
4397708	Lactobacillaceae	Unknown	Unknown
173209	Bacillaceae	Unknown	Unknown
146935	Lactobacillaceae	Lactobacillus	Unknown
780788	Xanthomonadaceae	Unknown	Unknown
69980	Aerococcaceae	Unknown	Unknown
4423201	Sphingobacteriaceae	Sphingobacterium	Multivorum
155345	Bacillaceae	Unknown	Unknown
4379247	Lactobacillaceae	Lactobacillus	Unknown
1138448	Bacillaceae	Virgibacillus	Unknown

6.2 Subway Data

We returned to the subway dataset with $n = 124$ samples from Porto Santo Island (Portugal), Sacramento, and New York City, and $p = 898$ OTUs aggregated for microbiome species. A key aspect of the challenge is the construction of a microbiome fingerprint that allows the identification of the geographical origin of a sample. For this reason, we relied on the Gower distance because of its accuracy in sample clustering applications, e.g., see the results in Table 3. Due to the large number of OTUs, the Skinny Bayesian approach was applied to significantly reduce the computational burden of the MCMC procedure.

Figure 5 displays the biplot for the top $m = 5$ OTUs. The cities are marked with symbols. Symbol “+” represents New York City, “/” represents Porto Santo Island, and “\$” represents Sacramento. The ordination of the data using the scores of the first two PCs appears to be successful at separating the cities. The arrows in the biplot correspond to the top $m = 5$ OTUs. The arrow labels are the abbreviated names for the species *Mycobacterium vacca*, *Staphylococcus equorum*, *Corynebacterium pilosum*, *Azospirillum spp*, and *Prevotella nanceiensis*.

The $n = 124$ samples were split into training and test cases in a 9:1 ratio. Based on 500 independent random splits, and using a similar classification strategy as for the Tobacco dataset, the test error rate for Gower distances was estimated to

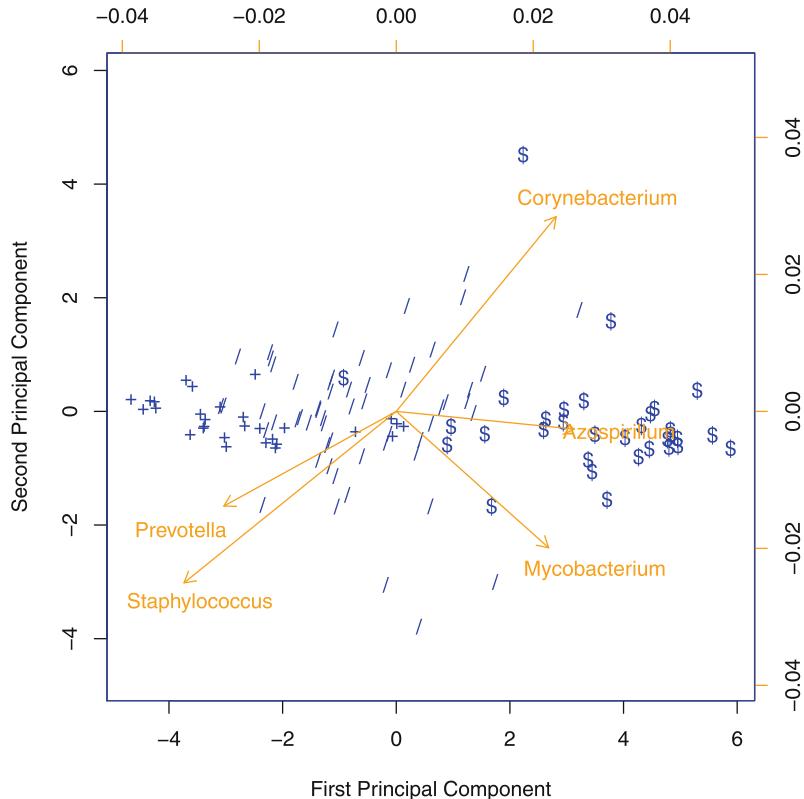


Fig. 5 Biplot for the subway dataset. The arrows represent the top 5 OTUs. Symbol “plus” represents New York City, “slash” represents Porto Santo Island, and “dollar” represents Sacramento

be 10.26% with a standard error of 0.37%. In contrast, covariance distances had a significantly greater test error rate of 13.37% with a standard error of 0.42%. Similar to the Tobacco dataset, these results demonstrate the advantages of alternative distance matrices.

7 Data Acknowledgement

The analyses presented in Sect. 6.2 are based on a subset of the raw metagenomics data provided by the MetaSUB International Consortium and the Scientific Programme Committee of the conference on Critical Assessment of Massive Data Analysis (CAMDA 2018). The primary data along with other supplementary data were publicly available on the challenge’s website as part of a forensic metagenomics challenge involving multiple international cities. A more recent extraction

of the CAMDA data has been analyzed in the chapter by Anyaso-Samuel et al. using supervised learning techniques.

8 Discussion

This chapter implements an approximate decomposition of abundance matrix X to restore, via the Bayesian paradigm, the duality between the orthonormal vectors associated with an arbitrary distance matrix Δ between the sample units and the orthonormal vectors of the OTUs. Although other researchers have proposed different types of optimization-based decomposition, there is no guarantee that the discovered decompositions converge to the global rather than a local minimum of the objective functions. In contrast, the Bayesian approach provides robust inference that relies on the posterior mean as the optimal estimate of unknown parameters and provides point estimates, standard errors, and credible intervals.

The effectiveness of the proposed Bayesian methodologies has been demonstrated via simulation studies. For the motivating Tobacco dataset, scree plots were utilized to identify the top 20 OTUs having the largest contributions to the variability of abundance matrix X . In this regard, the results of the two proposed Bayesian methods were qualitatively identical. The top two OTUs were found to be biologically distant but more closely related to the remaining OTUs in the set.

For the subway dataset, a biplot revealed that the ordination of the data using Gower distances was successful at separating the three cities from which the samples were taken. A quantitative analysis demonstrated that Gower distances outperform covariances with respect to classification of the geographical origin of a sample from the microbiome fingerprint based on the first two PCs. A similar result was observed for the Tobacco data, where Gower distances outperformed UniFrac distances and covariance matrices.

As future research, the approach is amenable to generalizations such as the incorporation of covariates. For an arbitrary distance matrix, this extended analytical framework may be used to make inferences about group differences in the approximate decomposition of the abundance matrix. These generalizations will be pursued elsewhere.

Supplementary Materials

R code implementing our procedure is available at <https://github.com/sguha-lab/Microbiome-SVD>.

Acknowledgments This work was partially supported by the National Science Foundation under Award DMS-1854003 to SG and by National Institutes of Health grant 5R03DE025625-02 to SD.

Appendix

Proof of Lemma 1

The $n \times n$ matrix, $\tilde{\mathbf{B}} = [\mathbf{B} : \mathbf{B}_0]$, is orthonormal and of full rank. Consequently,

$$\begin{aligned}
\mathcal{F}(V, \mathbf{D}) &= \|X - \mathbf{B}\mathbf{D}V^T\|_F^2 \\
&= \|\tilde{\mathbf{B}}^T X - \tilde{\mathbf{B}}^T \mathbf{B}\mathbf{D}V^T\|_F^2 \\
&= \left\| \begin{bmatrix} \mathbf{B}^T X \\ \mathbf{B}_0^T X \end{bmatrix} - \begin{bmatrix} \mathbf{B}^T \mathbf{B} \\ \mathbf{B}_0^T \mathbf{B} \end{bmatrix} \mathbf{D}V^T \right\|_F^2 \quad \text{since } \tilde{\mathbf{B}} \text{ is full-rank orthonormal} \\
&= \left\| \begin{bmatrix} \mathbf{B}^T X \\ \mathbf{B}_0^T X \end{bmatrix} - \begin{bmatrix} \mathbf{I}_q \\ \mathbf{0} \end{bmatrix} \mathbf{D}V^T \right\|_F^2 \quad \text{where } \mathbf{I}_q \text{ is the } q\text{-dimensional identity matrix} \\
&= \|\mathbf{B}^T X - \mathbf{D}V^T\|_F^2 + \|\mathbf{B}_0^T X\|_F^2.
\end{aligned}$$

Furthermore,

$$\begin{aligned}
\|\mathbf{B}^T X - \mathbf{D}V^T\|_F^2 &= \|VD - X^T \mathbf{B}\|_F^2 \\
&= \|VD - Q\|_F^2 \\
&= \text{tr}[(DV^T - Q^T)(VD - Q)] \\
&= \text{tr}(DV^T VD - Q^T VD - DV^T Q + Q^T Q) \\
&= \text{tr}(\mathbf{D}^2) - 2\text{tr}(Q^T VD) + \text{tr}(Q^T Q), \quad \text{since matrix } V \text{ is orthonormal} \\
&= \sum_{j=1}^q d_j^2 - 2\text{tr}(\mathbf{D} Q^T V) + \text{tr}(Q^T Q).
\end{aligned}$$

References

1. Caporaso, J., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F., Costello, E., et al.: QIIME allows analysis of high-throughput community sequencing data. *Nat Meth.* **7**(5), 335–336 (2010)
2. Downs, T.D.: Orientation statistics. *Biometrika* **59**, 665–676 (1972)
3. Hoff, P.D.: Simulation of the matrix Bingham–von Mises–Fisher distribution, with applications to multivariate and relational data. *J. Comput. Graph. Stat.* **18**(2), 438–456 (2009). <https://doi.org/10.1198/jcgs.2009.07177>
4. Kuczynski, J., Liu, Z., Lozupone, C., McDonald, D., Fierer, N., Knight, R.: Microbial community resemblance methods differ in their ability to detect biologically relevant patterns. *Nat. Methods* **7**, 813–819 (2010)

5. Legendre, P., Legendre, L.: Numerical Ecology, 3rd edn. Elsevier, Amsterdam (2012)
6. Rigon, T., Herring, A.H., Dunson, D.B.: A generalized Bayes framework for probabilistic clustering (2020). Preprint. arXiv:2006.05451
7. Satten, G.A., Tyx, R.E., Rivera, A.J., Stanfill, S.: Restoring the duality between principal components of a distance matrix and linear combinations of predictors, with application to studies of the microbiome. *PLoS One* **12**(1), e0168131, 1–15 (2017). <https://doi.org/10.1371/journal.pone.0168131>
8. Schloss, P.D., Westcott, S.L., Ryabin, T., Hall, J.R., Hartmann, M., Hollister, E.B., Weber, C.F.: Introducing Mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.* **75**(23), 7537–7541 (2009)
9. Tyx, R., Stanfill, S., Keong, L., Rivera, A., Satten, G., Watson, C.H.: Characterization of bacterial communities in selected smokeless tobacco products. *PLoS One* **11**(1), 7537–7541 (2016)

Part V

Special Topics

Tree Variable Selection for Paired Case–Control Studies with Application to Microbiome Data



Min Lu and Hemant Ishwaran

1 Introduction

Paired samples occur in microbiome studies when they are collected from different locations of the same individual or from paired individuals with familial ties. Human microbiome can be shared among family members with variations in each individual's microbial community [16, 18]. Suppose an identifiable “core microbiome” exists at the microbial gene level and deviations from this core are associated with different physiologic states. It is of interest to study how family ties play a role in these deviations. For example, if deviations from a core gut microbiome are associated with body mass index (BMI), we can define “individual” and “family” outcomes with labels obese/lean, where for example obese family means the individual comes from a family containing at least one member who is obese, and lean family means the individual comes from a family whose members are all lean. By analyzing such outcomes, we can examine how each array of microbial genes is associated with obesity both at the family and individual levels.

There are methods available for paired case–control studies, but they have limitations for analyzing the type of data considered here. For example, McNemar's test [11] is a statistical test used for paired data. However, it is primarily intended only for dichotomous features. Multilevel models with binary-dependent variables [1, 5] are another class of methods used. However, microbiome data is often high-dimensional, which makes it challenging to implement these approaches. Another challenge is that these models assume linearity but the association between obesity and microbiome features is likely to be nonlinear.

M. Lu · H. Ishwaran (✉)

Division of Biostatistics, University of Miami, Coral Gables, FL, USA

To illustrate our proposed methodology, we will use data from a cross-sectional study focusing on obesity in twins [9, 16, 17]. Data was collected from human stools of adult female monozygotic or dizygotic twins or their mothers. We utilize 142 of these samples with 54 pairs where pair is defined as family members including mother and daughters. The bacterial lineages present in the fecal microbiota of these individuals were characterized by 16S rRNA sequencing, targeting the full-length gene with an ABI 3730xl capillary sequencer. Sequences were identified by assignment to taxonomic groups using operational taxonomic units (OTUs). Specific details of how data was processed can be found in [16].

The original analysis found that obesity is associated with phylum-level changes in the microbiota and reduced bacterial diversity using linear approaches, such as PCA (principal components analysis). Here, we will focus on detecting which taxonomic groups are the most informative for obesity risk at both the family and individual levels using a novel approach that draws upon tree-based concepts.

2 Gini Index

Consider Y a categorical (factor) outcome such that $Y \in \{1, \dots, J\}$ for $J \geq 2$. Given a p -dimensional feature \mathbf{X} , the goal is to classify \mathbf{X} into one of the J classes. We call this J -class problem and $\{1, \dots, J\}$ the J -class labels for Y . Gini index is widely used for constructing classification trees that are nonparametric estimators used for the J -class problem. Given an input feature, classification trees work by identifying the unique leaf (terminal node) of the tree that \mathbf{X} resides within. Each leaf of the tree is labeled with a class among $\{1, \dots, J\}$ or a probability distribution over the classes, signifying that the leaf has been classified into either a specific class or a particular probability distribution, and this information is used for making decisions about \mathbf{X} .

A classification tree is built by splitting the data, constituting the root node of the tree, into subsets that constitute the successor children. The splitting is applied to features and is based on a pre-chosen splitting rule, which in addition to the Gini index includes splitting methods utilizing AUC (area under the ROC curve) and entropy metrics [3, 12]. This process is repeated on each derived subset in a recursive manner. The recursion is stopped when the subset at a node has all the same values of the outcome, or when a prespecified criterion is reached (such as minimal size of a node). The final nodes of the tree are referred to as the leaves. Classification trees can be combined to form ensemble estimators. An example is random forests [2], a popular tree-based learning method capable of handling a large number of predictors. In order to handle big data, rather than using a classification tree as just described, random forests is constructed by using random trees where each tree is constructed from subsampled data and where tree splitting employs random feature selection [2].

Classification tree splitting based on the Gini index splitting rule can be formally described as follows. If $\mathbf{p} = (p_1, \dots, p_J)$ are the data class proportions of Y for classes 1 through J , respectively, the Gini index of impurity is defined as

$$\phi(\mathbf{p}) = \sum_{j=1}^J p_j(1 - p_j) = 1 - \sum_{j=1}^J p_j^2.$$

As mentioned, classification trees are grown using the Gini index by splitting features recursively into left and right daughter nodes. In particular, tree splits are obtained by minimizing tree impurity. The Gini index split statistic for a split s on a continuous feature x_m at a given tree node is

$$\theta(Y, x_m, s) = \frac{n_l}{n} \phi(\mathbf{p}_l) + \frac{n_r}{n} \phi(\mathbf{p}_r),$$

where the subscripts $l = \{x_m \leq s\}$ and $r = \{x_m > s\}$ denote the left and right daughter nodes formed by the split on x_m at s (n_l and n_r are the sample sizes of the two daughter nodes where $n = n_l + n_r$ is the parent sample size). To reduce tree impurity, the goal is to find x_m and s to *minimize*

$$\theta(Y, x_m, s) = \frac{n_l}{n} \left(1 - \sum_{j=1}^J \frac{n_{j,l}^2}{n_l^2} \right) + \frac{n_r}{n} \left(1 - \sum_{j=1}^J \frac{n_{j,r}^2}{n_r^2} \right),$$

where $n_{j,l}$ and $n_{j,r}$ are the number of cases of class j in the left and right daughters, respectively, and $n_j = n_{j,l} + n_{j,r}$ are the number of cases of class j and $n = \sum_{j=1}^J n_j$. With some algebra, it can be shown this is equivalent to *maximizing* the split statistic

$$g(Y, x_m, s) = \frac{1}{n} \sum_{j=1}^J \frac{n_{j,l}^2}{n_l} + \frac{1}{n} \sum_{j=1}^J \frac{(n_j - n_{j,l})^2}{n - n_l}.$$

Although the Gini index is primarily used as a splitting rule, we observe that it can be used as a fast preliminary variable ranking method. This is because variables that are used to split a tree are often those variables that have highest variable importance as measured by prediction error, especially if these splits occur high up in the tree, i.e., near the root node that comprised all the data [8]. Thus, it is reasonable to rank variables in terms of size of their Gini index values calculated using the full data as this will generally rank variables by predictive power. For each of the p predictors x_1, \dots, x_p , define

$$G(Y, x_m) = g(Y, x_m, s_{\max}),$$

where

$$s_{\max} = \arg \max_s g(Y, x_m, s)$$

and $g(Y, x_m, s)$ is the split statistic calculated from the root node data (i.e., using the full data; thus, n is the sample size). Variables are ranked in order of importance

by size of $G(Y, x_m)$. This variable selection procedure is fully nonparametric and can be computed quickly even in big data settings. The following section provides a demonstration of how this approach works for our problem.

2.1 Simulation Analysis

Consider a binary class setting and denote the outcome as $Y^I \in \{0, 1\}$, where $Y^I = 0$ represents a lean individual and $Y^I = 1$ an obese individual. Family outcome is denoted as $Y^F \in \{0, 1\}$, where $Y^F = 0$ signifies an individual from a family with all lean members and $Y^F = 1$ indicates an individual from a family where at least one member is obese. Association with $Y^I = 1$ reflects how host adiposity influences the gut microbiome, whereas association with $Y^F = 1$ reflects environmental exposure influences. How the host genotype affects the gut microbiome under environmental exposure is reflected by an association with both $Y^I = 1$ and $Y^F = 1$.

We use the following simulation where Y^F is specified according to

$$\mathbb{P}\{Y^F = 1 | \mathbf{X} = \mathbf{x}\} = \text{logistic}(-2 + x_1 + x_2 + x_3 + 2 \times \mathbf{1}_{\{x_1 < 0.5\}}) \quad (1)$$

and Y^I is specified by

$$\mathbb{P}\{Y^I = 1 | Y^F = 1, \mathbf{X} = \mathbf{x}\} = \text{logistic}(-2 + x_4 + x_5 + x_6 + 2 \times \mathbf{1}_{\{x_4 < 0.5\}}), \quad (2)$$

where $\text{logistic}(\alpha) = 1/(1 + e^{-\alpha})$. In this scenario, x_1 , x_2 , and x_3 are associated with environmental exposures that cause the presence of obesity, while x_4 , x_5 , and x_6 are associated with host adiposity, given that the host is under these types of environmental exposures.

The feature space dimension was set to $p = 10$. Features were independently drawn from a uniform distribution $U(0, 1)$. Variables unrelated to outcome, representing noise variables, were also added to the design matrix. For Y^F , noise variables were x_4, \dots, x_{10} . For Y^I , noise variables were x_1, x_2, x_3 and x_7, \dots, x_{10} . Split statistics, $g(Y, x_m, s)$, are plotted in Fig. 1 for features x_1 , x_4 , and x_{10} and for both outcomes $Y = Y^F$ and $Y = Y^I$. Red color represents the family-level outcome Y^F , and blue is used for the individual-level outcome Y^I . Variable x_1 in (a) predicts obesity at the family level and is associated with Y^F , and the true optimal split point occurs at 0.5. We can see that the split statistic of x_1 is high for both Y^F and Y^I and both peak at around 0.5. Variable x_4 in (b) is associated with $\mathbb{P}\{Y^I = 1 | Y^F = 1\}$, therefore is associated with Y^I , and has a true optimal split point of 0.5. We can see that the split statistic $g(Y^I, x_4, s)$ is high for Y^I and reaches its peak near 0.5 (although not exactly at the true value—we will come back to this point later). In contrast, the split statistic $g(Y^F, x_4, s)$ for Y^F does not at all have an optimized value near 0.5 and its peak value occurs near its edge. This edge effect is typical of noisy variables and is a property of the Gini splitting rule called end-cut preference, ECP [6]. Variable x_{10} in (c) is a noise variable, and its split statistic is low for

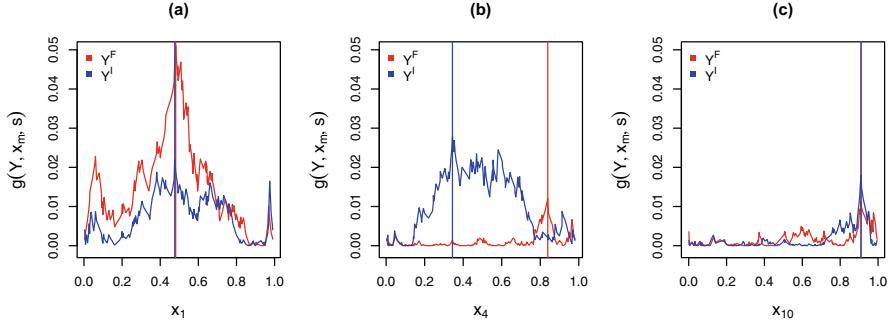


Fig. 1 Univariate split statistics for x_1 , x_4 , and x_{10} from simulation (1)–(2). Values $g(Y, x_m, s)$ are shown across different split values s . Red and blue display family-level outcome Y^F and individual-level outcome Y^I , respectively. Vertical lines mark the optimal split statistic $G(Y, x_m)$. Variable x_1 is associated with Y^F with true optimal split point of 0.5. Variable x_4 is associated with $\mathbb{P}\{Y^I = 1 | Y^F = 1\}$ with true optimal split point of 0.5. Variable x_{10} is a noise variable

both Y^F and Y^I . Observe that its optimal split points are close to the edge for both outcomes, which as stated is typical behavior of a noisy variable.

Comparing the results across Fig. 1, it is clear that $G(Y, x_m)$, which is the highest point of $g(Y, x_m, s)$, is useful for variable ranking. However, focusing only on family-level outcomes (red color) will ignore features like x_4 that are related to the individual-level outcome (blue color). Checking both split statistics clearly helps better understand the underlying associations.

3 Multivariate Gini Index

Tang and Ishwaran [13] defined a multivariate Gini index split statistic obtained by averaging univariate Gini split statistics. For the bivariate outcome problem, this can be described as

$$\bar{g}_u(Y^F, Y^I, x_m, s) = \frac{1}{2} [g(Y^F, x_m, s) + g(Y^I, x_m, s)].$$

The subscript “ u ” is used to emphasize that the split statistic is unweighted. We can define

$$\overline{G}_u(Y^F, Y^I, x_m) = \bar{g}_u(Y^F, Y^I, x_m, s_{u_{\max}})$$

for ranking variables, where

$$s_{u_{\max}} = \arg \max_s \overline{g}_u(Y^F, Y^I, x_m, s).$$

Larger values of $\bar{G}_u(Y^F, Y^I, x_m)$ identify informative variables and smaller values indicate noise variables.

3.1 Conditional Gini Index

The problem with the split statistic $\bar{g}_u(Y^F, Y^I, x_m, s)$ is that by averaging across the outcomes it ignores the correlation between Y^F and Y^I . To resolve this issue, we introduce the following conditional Gini split statistic.

Let $\pi_c = \mathbb{P}\{Y^I = 1 | Y^F = 1\}$ be the population proportion of obese cases among individuals with at least one obese family member. The subscript “ c ” is used to emphasize this is a conditional probability. Because there are only two classes, we have $\mathbf{p}_c = (p_c, 1 - p_c)$ and $\phi(\mathbf{p}_c) = 2p_c(1 - p_c)$, where p_c is the sample estimator of π_c . For a split s on variable x_m , the conditional Gini split statistic is defined as

$$\theta_c(Y^F, Y^I, x_m, s) = \frac{\tilde{n}_l}{\tilde{n}}\phi(\mathbf{p}_c) + \frac{\tilde{n}_r}{\tilde{n}}\phi(\mathbf{p}_c),$$

where as before subscripts l and r denote the left and right daughter nodes formed by the split. The numbers of cases $Y^F = 1$ in the daughters are \tilde{n}_l and \tilde{n}_r , where $\tilde{n} = \tilde{n}_l + \tilde{n}_r$. The numbers of these cases where $Y^I = 1$ in the left and right daughters are denoted by $\tilde{n}_{1,l}$ and $\tilde{n}_{1,r}$ respectively. It can be shown that minimizing $\theta_c(Y^F, Y^I, x_m, s)$ is equivalent to maximizing

$$g_c(Y^F, Y^I, x_m, s) = \frac{\tilde{n}_{1,l}^2}{\tilde{n}\tilde{n}_l} + \frac{\tilde{n}_{1,r}^2}{\tilde{n}\tilde{n}_r}.$$

We can define

$$G_c(Y^F, Y^I, x_m) = g_c(Y^F, Y^I, x_m, s_{c_{\max}})$$

for ranking variables, where $s_{c_{\max}} = \arg \max_s g_c$.

Now because $g_c(Y^F, Y^I, x_m, s)$ conditions on $Y^F = 1$, it is not designed to identify signal affecting Y^F . To resolve this, define the conditional weighted split statistic

$$\bar{g}_{cw}(Y^F, Y^I, x_m, s) = \frac{1}{w_F + w_I} [w_F \cdot g(Y^F, x_m, s) + w_I \cdot g_c(Y^F, Y^I, x_m, s)]$$

for detecting features that affect both Y^F and Y^I . Observe that when $w_F = w_I = 1$, this becomes an unweighted split statistic and will be denoted by $\bar{g}_{cu}(Y^F, Y^I, x_m, s)$.

Weighted indices can be calculated as $w_F = \sum_i^n \mathbf{1}_{\{Y_i^F=1\}}$ and $w_I = \sum_i^n \mathbf{1}_{\{Y_i^I=1\}}$, which adjust for the fact that there are always more obese cases for Y^F than Y^I . The maximum value for the conditional weighted split statistic is

$$\bar{G}_{cw}(Y^F, Y^I, x_m) = \bar{g}_{cw}(Y^F, Y^I, x_m, s_{cw_{\max}}),$$

where $s_{cw_{\max}} = \arg \max_s \bar{g}_{cw}$. In a likewise fashion, define the maximum conditional unweighted split statistic $\bar{G}_{cu}(Y^F, Y^I, x_m)$.

Figure 2 displays: (a) $g_c(Y^F, Y^I, x_m, s)$, (b) $\bar{g}_u(Y^F, Y^I, x_m, s)$, (c) $\bar{g}_{cu}(Y^F, Y^I, x_m, s)$, and (d) $\bar{g}_{cw}(Y^F, Y^I, x_m, s)$ for variables x_1, x_4 , and x_{10} from the simulation (1)–(2). Variable x_4 affects the conditional probability $\mathbb{P}(Y^I = 1 | Y^F = 1)$, which is plotted in purple color. Returning to the point made earlier regarding Fig. 1b, when comparing Fig. 2a to Fig. 1b, we find $g_c(Y^F, Y^I, x_4, s)$ characterizes x_4 better than $g(Y^I, x_4, s)$ as the maximum value is closer to the true splitting point 0.5. Another point to observe is that the goal of $\bar{g}_u(Y^F, Y^I, x_m, s)$ and $\bar{g}_{cu}(Y^F, Y^I, x_m, s)$ is to detect features associated with Y^F and/or Y^I . However, $\bar{g}_u(Y^F, Y^I, x_m, s)$ in (b) is less effective than $\bar{g}_{cu}(Y^F, Y^I, x_m, s)$ in (c) because it ranks x_4 similarly to noise variable x_{10} (shown in orange). In contrast, $\bar{g}_{cu}(Y^F, Y^I, x_m, s)$ in (c) and the weighted $\bar{g}_{cw}(Y^F, Y^I, x_m, s)$ in (d) properly rank x_4 as more informative than x_{10} . In fact, the weighted split statistic tends to do an even better job.

Figure 3 displays maximum Gini split statistics for all $p = 10$ variables averaged over 100 independent replications. For convenient calibration, the averaged split statistic for the noise variable x_7 is used as a selection cutoff. When comparing subfigures (c) with (b), we see that $G_c(Y^F, Y^I, x_m)$ performs better in terms of selecting the true signals, x_4, x_5 , and x_6 , than $G(Y^I, x_m)$. When comparing subfigures (f) with (d), we observe that the weighted Gini split statistic utilizing the conditional Gini index, $\bar{G}_{cw}(Y^F, Y^I, x_m)$, outperforms the simple averaged Gini split statistic, $\bar{G}_u(Y^F, Y^I, x_m)$, in selecting the true signal variables x_1, \dots, x_6 (in (d) the informative variable x_6 is not selected, whereas the noise variable x_{10} is selected). The performances of $\bar{G}_{cu}(Y^F, Y^I, x_m)$ and $\bar{G}_{cw}(Y^F, Y^I, x_m)$ are roughly similar except that noise variable x_{10} is less likely to be chosen using $\bar{G}_{cw}(Y^F, Y^I, x_m)$. Thus as before, the weighted split statistic tends to do a better job. Finally, when comparing subpanel (f) to (a) notice that $\bar{G}_{cw}(Y^F, Y^I, x_m)$ is as good as $G(Y^F, x_m)$ in identifying variables x_1, x_2, x_3 related to Y^F . However, this does not mean $G(Y^F, x_m)$ is not useful, since when combined with $\bar{G}_{cw}(Y^F, Y^I, x_m)$ it allows one to detangle variable relationships with the two outcomes.

4 Variable Importance

Another effective tool for variable selection is variable importance (VIMP). The permutation VIMP for a variable x_m is the prediction error for the model sub-

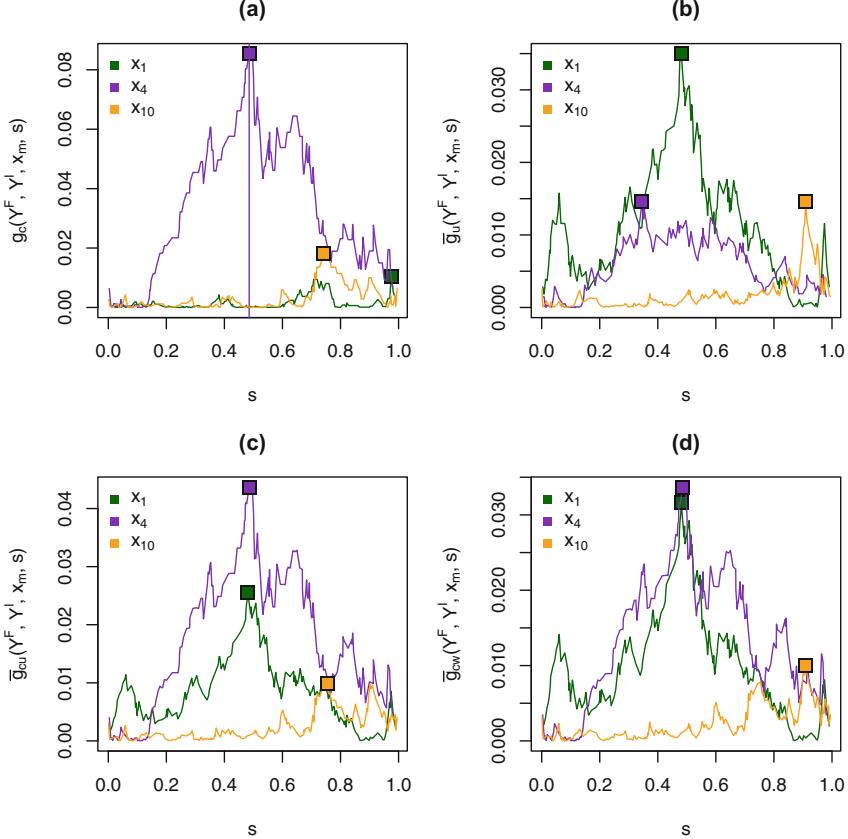


Fig. 2 Multivariate split statistics for x_1 , x_4 , and x_{10} from simulation (1)–(2). Curves displayed are: (a) $g_c(Y^F, Y^I, x_m, s)$, (b) $\bar{g}_u(Y^F, Y^I, x_m, s)$, (c) $\bar{g}_{cu}(Y^F, Y^I, x_m, s)$, and (d) $\bar{g}_{kw}(Y^F, Y^I, x_m, s)$ with maximum statistic marked by a square point

tracted from the prediction error for the model using data that randomly permutes x_m [7]. This procedure can be implemented over independent bootstrap samples and the value averaged to obtain a more stable estimator [7]. More formally, let $\hat{PE}(Y)$ be the averaged out-of-sample (called out-of-bag and abbreviated as OOB) misclassification error for the original model. Let $\hat{PE}(Y, x_m^*)$ be the averaged OOB misclassification error when x_m is randomly permuted. The VIMP for x_m is

$$I(Y, x_m) = \hat{PE}(Y, x_m^*) - \hat{PE}(Y).$$

To determine if variables affect the conditional probability $\mathbb{P}(Y^I = 1 | Y^F = 1)$, we define a conditional VIMP analogous to the conditional Gini index. Conditional VIMP is calculated by restricting the data to those cases where $Y^F = 1$. The conditional VIMP index for x_m is

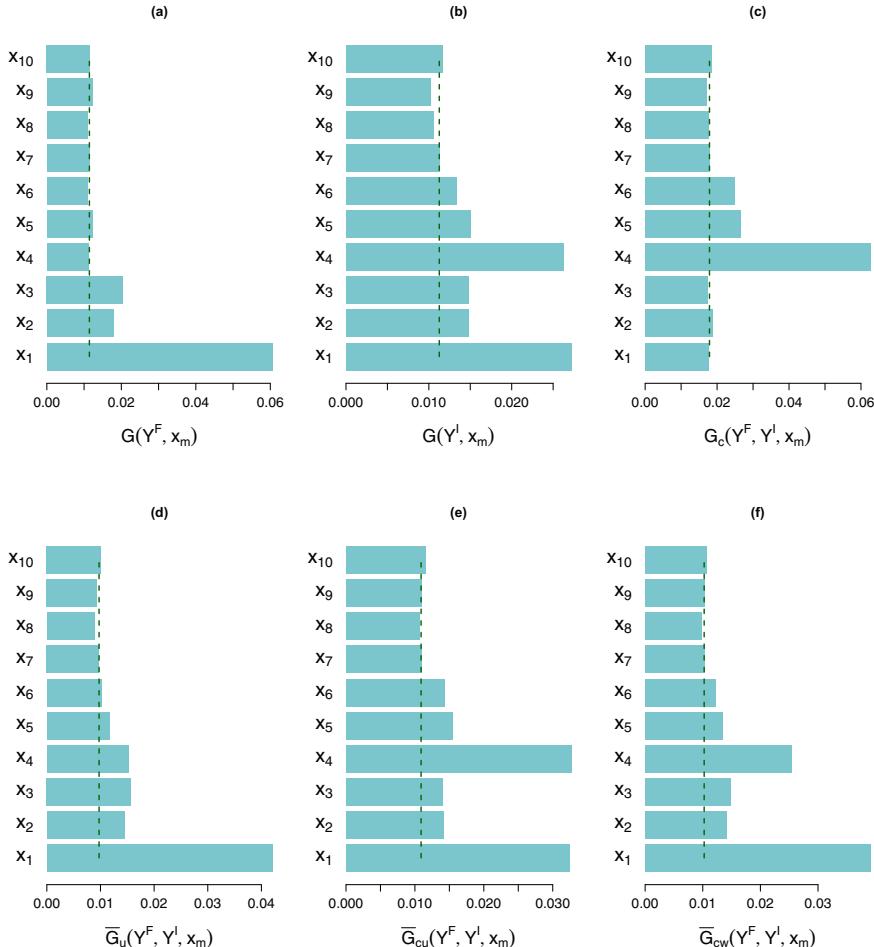


Fig. 3 Variable ranking from maximum split statistics for simulation (1)–(2) repeated 100 times independently. Dashed line is the averaged value of maximum Gini split statistic for noise variable x_7 that represents a convenient cut-off value

$$I_c(Y^I, x_m) = \hat{\text{PE}}_c(Y^I, x_m^*) - \hat{\text{PE}}_c(Y^I).$$

Figure 4 displays VIMP for all p features for our simulation. Values have been averaged over 100 independent replications. Unconditional VIMP, $I(Y^F, x_m)$, for Y^F displayed in subpanel (a) successfully ranks the true signal variables x_1, x_2 and x_3 as the most informative. When comparing subpanel (c) to (b), we see that conditional VIMP, $I_c(Y^I, x_m)$, is better at selecting true signal variables x_4, x_5 , and x_6 than unconditional VIMP, $I(Y^I, x_m)$. In subfigure (b), VIMP for x_1 is very large and would lead to incorrect selection compared with (c).

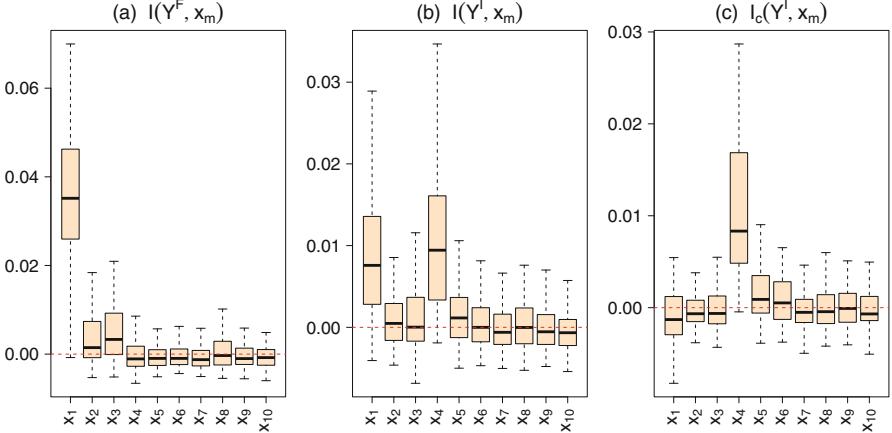


Fig. 4 Variable importance from simulation (1)–(2) averaged over 100 independent replications

5 Analysis of Obesity Using Microbiome Data

Now we return to the microbiome obesity data described earlier ($n = 142$ and $p = 174$). Outcomes were coded as before: $Y^I = 0$ represents a lean individual, $Y^I = 1$ an obese individual, $Y^F = 0$ signifies an individual from a family with all lean members, and $Y^F = 1$ indicates an individual from a family where at least one member is obese. Table 1 of the Appendix provides convenient abbreviated names for features. The features were originally coded using names of kingdom, phylum, class, order, family, genus, and species separated with a dash line and capital letter [9]. In order to make the name shorter for the figures, we use Table 1 to shorten the name for some classification labels. However to avoid duplication, we did not shorten all names. Therefore, some x-labels in Fig. 5 use full classification names.

Figure 5 displays split statistics for 6 representative features, chosen to illustrate how host and environmental factors affect the gut microbiome. Univariate split statistics $g(Y^F, x_m, s)$ for the family outcome Y^F are shown using red lines, and conditional split statistics $g_c(Y^F, Y^I, x_m, s)$ are displayed using orange lines. Bivariate split statistics, $\bar{g}_{cu}(Y^F, Y^I, x_m, s)$ and $\bar{g}_{cw}(Y^F, Y^I, x_m, s)$, lie between these two lines. Recall when optimal split points appear toward the edge of feature's range that this is a sign of a noisy feature (referred to as the ECP property [6]).

Subfigures (a), (b), (c) represent features informative for the environmental outcome Y^F . In all three figures, $g(Y^F, x_m, s)$ takes large values across the range of feature values. However, these three features are not informative for $\mathbb{P}\{Y^I = 1|Y^F = 1\}$ as $g_c(Y^F, Y^I, x_m, s)$ is near zero in all instances. Thus, they do not reflect how host adiposity influences the gut microbiome under environmental exposure.

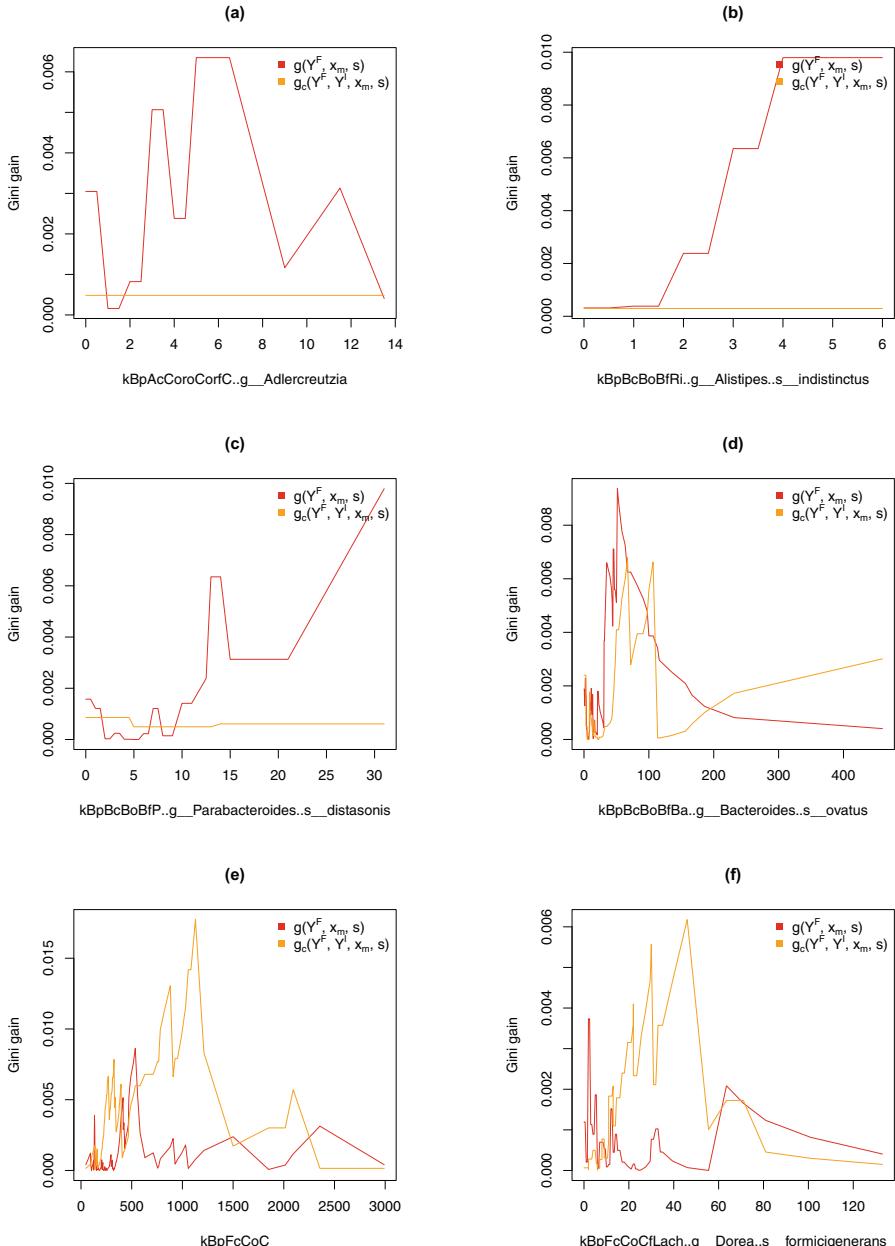


Fig. 5 Split statistics for microbiome obesity data. Shown are 6 representative variables illustrating how taxonomic groups predict obesity risk at the family level (shown using the univariate Gini split statistic on Y^F , $g(Y^F, x_m, c)$, plotted in red) and at the individual level (shown using the univariate conditional Gini split statistic $g_c(Y^F, Y^I, x_m, c)$, plotted in orange). Variable names are abbreviated according to Table 1

Subfigures (d) and (e) represent features that are informative for both $Y^F = 1$ and $\mathbb{P}\{Y^I = 1|Y^F = 1\}$ as both $g(Y^F, x_m, s)$ and $g_c(Y^F, Y^I, x_m, s)$ assume relatively large values. These features identify influences from both environmental exposure and host adiposity. For (d), the two maximum split statistics are nearly the same, which suggests that effect of environmental exposure and host adiposity is roughly the same for this feature. For (e), $g_c(Y^F, Y^I, x_m, s)$ attains a much larger maximum statistic than $g(Y^F, x_m, s)$ at a higher feature value. This suggest the effect of environmental exposure and host adiposity depends on the feature value, for example, whether the feature value is larger than 500 or 1000.

Subfigure (f) is a feature that mainly reflects the influence from host adiposity, rather than environmental exposure. This is because values of $g(Y^F, x_m, s)$ are overall small and its optimal split point is close to the edge of its range, signaling that it is likely a noisy variable for Y^F .

The values of $G(Y^F, x_m)$ and $G_c(Y^F, Y^I, x_m)$ are given in Fig. 6. The sizes of circles are scaled proportional to $\bar{G}_{cw}(Y^F, Y^I, x_m)$. Phylum groups are used to color circles. It is interesting to note that features informative for $Y^F = 1$ and $\mathbb{P}\{Y^I = 1|Y^F = 1\}$ belong primarily to the Fusobacteria phylum. Generally, the values of $G_c(Y^F, Y^I, x_m)$ are smaller than $G(Y^F, x_m)$. However, when they are

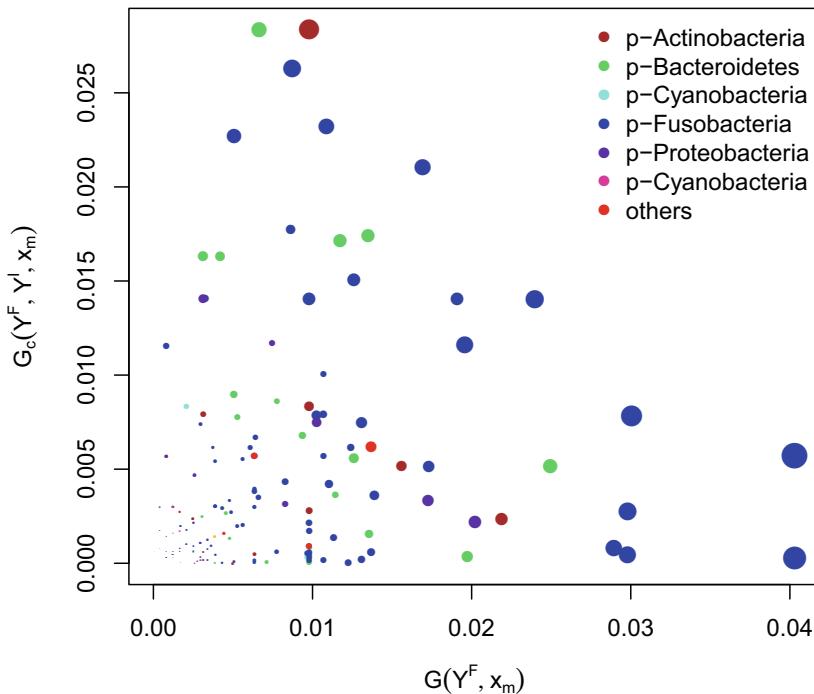


Fig. 6 Comparison between $G(Y^F, x_m)$ and $G_{Y^F=1}(Y^I, x_m)$. The sizes of circles are proportional to $\bar{G}_{cw}(Y^F, Y^I, x_m)$. The color of circles identifies the phylum

weighted to obtain $\overline{G}_{cw}(Y^F, Y^I, x_m)$, we can see that there is a nice balancing of values.

Finally, Fig. 7 displays unconditional VIMP, $I(Y^F, x_m)$, and conditional VIMP, $I_c(Y^I, x_m)$, for all p variables. Many variables have small or negative values, thus showing that VIMP can be used as an effective means to dimension. Note that due to invariance of trees under monotonic transformations of features, split statistics and VIMP are invariant to the magnitude of the feature values. For split statistics, it is the quantile of the split that makes a difference. Thus, we did not normalize independent variables, which were OTU count values [9].

Our results are consistent with previous studies. Turnbaugh et al. [15, 16] found that obesity is associated with phylum-level changes in the microbiota. We found the same as displayed in Fig. 6. Moreover, our findings (see Fig. 5) have shown that changes occur differently at the family and individual levels. A similar finding was observed in a study of high-fat diet-induced obesity and diabetes in mice [4]. This study found a high-fat diet-altered proportion of *Bacteroides*-related bacteria and reduced *Bifidobacteria* (Gram-positive, phylum Actinobacteria).

We carefully note that our analysis can only establish an association between microbiome and obesity [14] and not causation. Casual inference is far more demanding, and further studies would be needed to be able to move beyond associative analyses [10].

6 Discussion

Fast nonparametric selection of features that accounts for correlation in paired data is a valuable tool for microbiome data analysis. Variable selection procedures can choose features that reflect influences from external effects (between pairs) and internal effects (within pairs), but without taking into account the paired structure of the data, they will be inefficient in separating the two types of effects. Our proposed conditional Gini split statistic, when used alone or averaged with univariate Gini split statistics, serves two purposes. First, the maximum value of the split statistic can be used for variable ranking and variable selection. Conditional Gini is able to select variables reflecting how the microbiome is affected by host adiposity given the same environmental exposures. Second, how the value of the split statistic varies within a feature provides useful insight into the magnitude of the external and/or internal effects. The optimal split point for conditional Gini represents the threshold that a feature can separate lean and obese individuals given the same environmental exposure. We demonstrated these two aspects in a systematic comparative simulation and through a real data application. We found that the paired structure of the data played a very strong role in performance of our methods. Without controlling for family level of obesity, features that only affect individual level of obesity are often noticeably masked.

There are other variable selection procedures designed for multivariate outcomes. However, in big data settings, computational speed plays a key role. Practically speaking, the best method is not always optimal for the researcher because computational times can be too long. Our Gini split statistics can be rapidly computed for a large number of features in big data settings, and because the calculations are univariate, the procedure could be parallelized to further reduce runtimes. Users can simulate a noise feature to determine the cutoff for screening noise variables. Potentially, our Gini indices can be used as tree splitting rules so that all the features can be taken into consideration together. Moreover, our approach could leverage powerful machine learning methods such as random forests and boosting to provide a direct approach to analyze paired data. Another potential improvement to our work would be to use additional data to study the effect of number of individuals on Y^F . In the analysis we used, all families have 2 or 3 individuals, which made it impossible for us to study the effect of number of individuals.

Acknowledgments This work was supported by the National Institutes of Health (grant numbers R01 CA200987 and R01 HL141892 to H.I.).

Appendix

See Table 1 and Fig. 7.

Table 1 Abbreviated feature names for microbiome obesity data

Abbrev.	Full form	Abbrev.	Full form
kB	k-Bacteria	oBi	..o-Bifidobacteriales
pF	..p-Firmicutes	oE	..o-Erysipelotrichales
pA	..p-Actinobacteria	oL	..o-Lactobacillales
pB	..p-Bacteroidetes	fB	..f-Bifidobacteriaceae
pC	..p-Cyanobacteria	fBa	..f-Bacteroidaceae
pF	..p-Fusobacteria	fC	..f-Coriobacteriaceae
pP	..p-Proteobacteria	fL	..f-Lactobacillaceae
pS	..p-Synergistetes	fLach	..f-Lachnospiraceae
cA	..c-Actinobacteria	fM	..f-Micrococcaceae
cB	..c-Bacteroidia	fP	..f-Porphyromonadaceae
cBci	..c-Bacilli	fPe	..f-Peptostreptococcaceae
cC	..c-Clostridia	fPr	..f-Prevellaceae
cCor	..c-Coriobacteriia	fR	..f-Ruminococcaceae
cE	..c-Erysipelotrichi	fRi	..f-Rikenellaceae
oA	..o-Actinomycetales	fS	..f-Streptococcaceae
oC	..o-Clostridiales	fV	..f-Veillonellaceae
oCor	..o-Coriobacteriales	gB	..g-Bifidobacterium
oB	..o-Bacteroidales	gC	..g-Corynebacterium

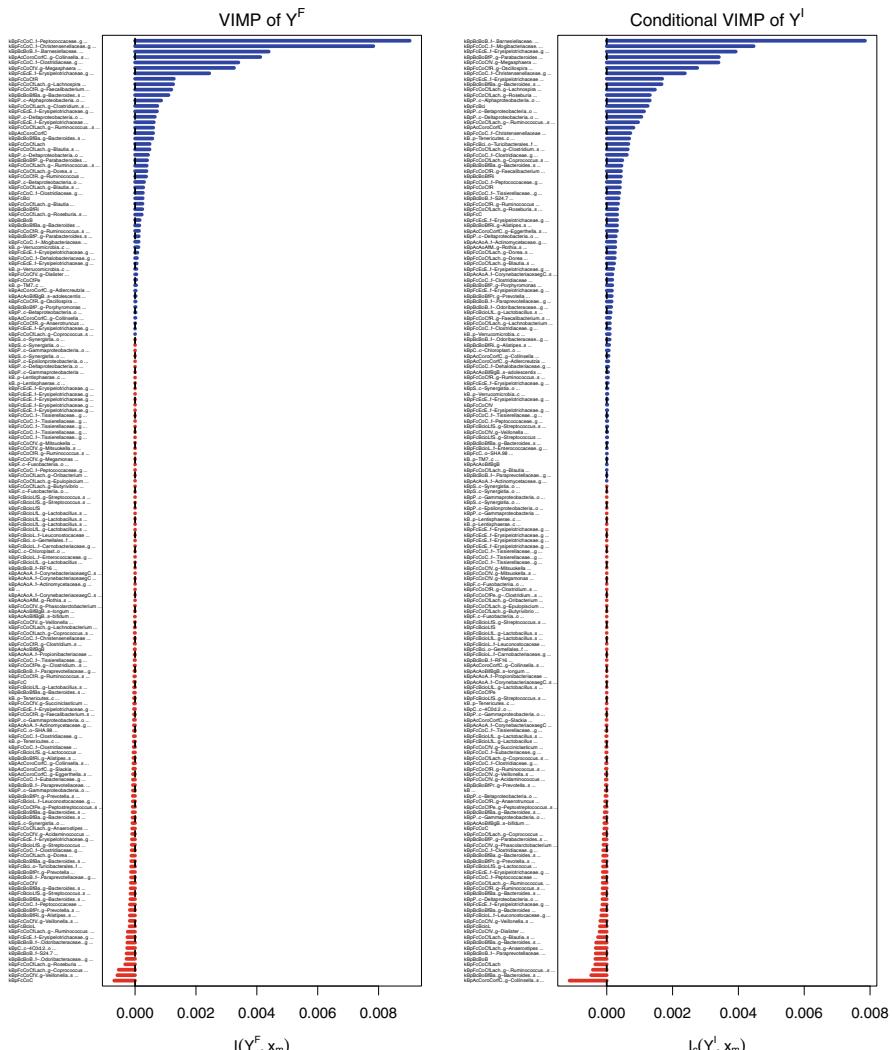


Fig. 7 Variable ranking using VIMP for microbiome data of obesity. Variables with higher value on the left reflect how the gut microbiome is influenced by environmental factors. Variables with higher values in the right reflect how gut microbiome is affected by host adiposity given the environmental exposures

References

1. Benedetti, A., Platt, R., Atherton, J.: Generalized linear mixed models for binary data: are matching results from penalized quasi-likelihood and numerical integration less biased? *PLoS One* **9**(1), e84601 (2014)
2. Breiman, L.: Random forests. *Machine Learning* **45**, 5–32 (2001)

3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Belmont, California (1984)
4. Cani, P.D., Bibiloni, R., Knauf, C., Waget, A., Neyrinck, A.M., Delzenne, N.M., Burcelin, R.: Changes in gut microbiota control metabolic endotoxemia-induced inflammation in high-fat diet-induced obesity and diabetes in mice. *Diabetes* **57**(6), 1470–1481 (2008)
5. Capanu, M., Gönen, M., Begg, C.B.: An assessment of estimation methods for generalized linear mixed models with binary outcomes. *Stat. Med.* **32**(26), 4550–4566 (2013)
6. Ishwaran, H.: The effect of splitting on random forests. *Machine Learning* **99**, 75–118 (2015)
7. Ishwaran, H., Lu, M.: Standard errors and confidence intervals for variable importance in random forest regression, classification, and survival. *Stat. Med.* **38**, 558–582 (2019)
8. Ishwaran, H., Kogalur, U.B., Gorodeski, E.Z., Minn, A.J., Lauer, M.S.: High-dimensional variable selection for survival data. *J. Am. Stat. Assoc.* **105**(489), 205–217 (2010)
9. Knights Lab: Monozygotic or dizygotic twin pairs concordant for BMI class, and their mothers (Project Turnbaugh 2009). <https://github.com/knights-lab/MLRepo/tree/master/datasets/turnbaugh>. Cited 28 Mar (2020)
10. Maruvada, P., Leone, V., Kaplan, L.M., Chang, E.B.: The human microbiome and obesity: moving beyond associations. *Cell Host Microbe* **22**(5), 589–599 (2017)
11. McNemar, Q.: Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **12**(2), 153–157 (1947)
12. Shalev-Shwartz, S., Ben-David, S.: Chapter 18. Decision Trees. *Understanding Machine Learning*. Cambridge University Press, Cambridge (2014)
13. Tang, F., Ishwaran, H.: Random forest missing data algorithms. *Stat. Anal. Data Min. ASA Data Sci. J.* **10**(6), 363–377 (2017)
14. Tsai, F., Coyle, W.J.: The microbiome and obesity: is obesity linked to our gut flora? *Curr. Gastroenterol. Rep.* **11**(4), 307–313 (2009)
15. Turnbaugh, P.J., Gordon, J.I.: The core gut microbiome, energy balance and obesity. *J. Physiol.* **587**(17), 4153–4158 (2009)
16. Turnbaugh, P.J., Hamady, M., Yatsunenko, T., Cantarel, B.L., Duncan, A., Ley, R.E., ... Egholm, M.: A core gut microbiome in obese and lean twins. *Nature* **457**(7228), 480–484 (2009)
17. Vangay, P., Hillmann, B. M., Knights, D.: Microbiome Learning Repo (ML Repo): A public repository of microbiome regression and classification tasks. *GigaScience* **8**(5), giz042 (2019)
18. Xia, Y., Sun, J., Chen, D.G.: Statistical Analysis of Microbiome Data with R. Springer, Singapore (2018)

Networks for Compositional Data



Jing Ma, Kun Yue, and Ali Shojaie

1 Introduction

Networks and network models are ubiquitously used in biomolecular research and systems biology to capture the interactions among components of biological systems [3]. By capturing interactions among genes, proteins, and metabolites, networks have led to invaluable insight into complex mechanisms governing cellular functions and biological phenotypes [4], as well as aberrations in these mechanisms in various complex diseases [79].

Microbes are known to interact with each other as members of microbial communities. These complex interactions, including mutualism, commensalism, parasitism, amensalism, and competition [54], can be compactly captured through microbial interaction networks. Thus, with the advent of high-throughput technologies, particularly 16s rRNA and metagenomics sequencing, the microbiome research community is increasingly interested in modeling, inferring, and analyzing microbial interaction networks.

The analysis of microbial interaction networks presents unique challenges. On the one hand, the existing knowledge of microbial interactions is incomplete, covering a small subset of possible interactions among hundreds of thousands of microbes identified in humans [24]. On the other hand, microbial communities and interactions are inherently dynamic [18], varying in different biological/physiological conditions and in different organisms/environments. Thus, to understand how

J. Ma

Fred Hutchinson Cancer Research Center, Seattle, WA, USA
e-mail: jingma@fredhutch.org

K. Yue · A. Shojaie (✉)

University of Washington, Seattle, WA, USA
e-mail: yuek@uw.edu; ashojaie@uw.edu

microbes interact with each other and how changes in these interactions may lead to various diseases/phenotypes, efficient and accurate methods for inferring microbial interaction networks from microbial abundance data are necessary.

Network inference is a fundamental but challenging problem in computational biology [62]. At its core, this problem is related to identifying associations among random variables. As such, the plethora of existing network inference procedures can be broadly categorized based on the types of association they consider. The simplest and most widely used approach is to infer biological interactions based on the (Pearson) correlation coefficient between each pair of variables [94]. Despite its simplicity, this approach has two key limitations: (1) correlation coefficient only captures linear associations between a pair of variables and is thus not appropriate for capturing complex nonlinear associations commonly observed in biology and (2) correlation coefficient may not distinguish between direct and indirect associations among variables. For instance, abundances of two microbes that are in mutualism relationships with a third microbe would likely be significantly correlated, even if the two microbes do not have a direct ecological relationship.

The first limitation of Pearson correlation can be (partially) mitigated by using rank-based correlation measures, such as Spearman correlation and Kendal's tau. As a more flexible alternative, semi- and non-parametric measures of associations, such as mutual information [59] and kernel-based tests of association [86], can be used to capture more complex interaction mechanisms; however, these flexible approaches require larger sample sizes and additional computation. The second limitation of Pearson correlation can also be addressed by considering conditional measures of association that aim to capture the dependence among each pair of variables *conditioned on other variables*. Estimating conditional associations is generally more challenging than estimating marginal associations. However, developments in regularized estimation [21] and graphical modeling [57] have led to significant advances in this area over the past decade [20].

The two classes of approaches described above, namely those based on marginal and conditional associations, have also been used to infer microbial interaction networks from high-throughput abundance data. However, due to challenges in normalization and sample preparation, microbial activities are often measured via relative abundances [49], wherein microbial measurements across each sample add to one. This *compositional* nature of microbiome data [34] leads to spurious (negative) correlations among taxa that render existing network inference procedures inaccurate. A number of recent procedures aim to overcome this challenge by developing tailor-made network estimation procedures for high-dimensional compositional data.

Recently proposed methods for inferring microbial networks utilize different strategies and focus on different aspects of the problem. Choosing the appropriate method requires a systematic comparison of their performances in different settings (e.g. with different sample sizes and a different number of microbes). Such comparisons are particularly important given the paucity of gold standards for microbial interactions and the challenges of experimentally validating inferred microbial interactions [15]. However, generating realistic microbiome data—i.e. relative

abundances corresponding to complex interaction mechanisms—is challenging. This challenge is compounded with the general difficulty of evaluating network inference procedures due to the variety of performance measures that can be used to compare two networks [74]. As such, systematic comparisons of existing methods are currently lacking.

In this chapter, we review commonly used procedures for inferring microbial interaction networks. This includes methods based on both marginal and conditional associations. We also present a comprehensive evaluation of these methods using two classes of simulated data sets: first, we evaluate whether each procedure can correctly identify a *null model* with no interactions. Second, we evaluate the performance of each method in detecting edges in simulated networks using data from two parametric models for compositional microbiome data. Together, these evaluations provide informative guidelines for practitioners on the appropriate choice of methods for inferring microbial networks.

The rest of this chapter is organized as follows. Section 2 reviews the methods for inferring microbial interaction networks, including well-known methods based on *marginal* and *conditional* associations among microbes. In Sect. 3, we present our data generation and simulation framework, including both null models, as well as *in silico* compositional data from simulated networks. Results of our simulation studies for comparing the existing procedures are presented in Sect. 4. We end with a discussion of future research directions in Sect. 5.

2 Methods

Table 1 provides an overview of the methods considered in this chapter for inferring microbial networks. ReBoot, SparCC, CCLasso, and COAT estimate the marginal association network via correlations, whereas SPIEC-EASI, gCoda, and SPRING estimate the conditional association network defined by partial correlations. Of all methods, only ReBoot and SparCC quantify the uncertainty of the inferred associations. All methods are implemented in R.

Table 1 Microbial network estimation methods considered in this chapter. CCLasso, COAT, and gCoda are available as R functions available at the authors’ GitHub pages

Method	Output	Error calls	R implementation	Reference
ReBoot	Correlation	Inference	ccrepe 1.20.0	[25]
SparCC	Correlation	Inference	SpiecEasi 1.1.0	[31]
CCLasso	Correlation	Estimation	huayingfang/CCLasso	[22]
COAT	Correlation	Estimation	yuanpeicao/COAT	[9]
SPIEC-EASI	Partial correlation	Estimation	SpiecEasi 1.1.0	[46]
gCoda	Partial correlation	Estimation	huayingfang/gCoda	[23]
SPRING	Partial correlation	Estimation	SPRING 1.0.3	[89]

2.1 Learning Networks from Marginal Associations

Let $\mathbf{W}_i = (W_{i,1}, \dots, W_{i,p})^\top$ be the unobserved counts of p taxa in one environmental sample, where the superscript \top denotes the transpose of a vector. Of key interest is the covariance $\Sigma^0 = \text{Var}(\log \mathbf{W}_i) = (\sigma_{j,k}^0)_{p \times p}$, which provides valuable insights into the ecological interactions among microbial taxa [15]. If the absolute abundances \mathbf{W}_i are available, then marginal association measures such as Pearson and Spearman correlations offer a convenient approach for defining microbial correlation networks. Despite efforts into quantifying the total microbial load from environmental samples [77, 78], absolute abundances are typically unknown. Instead, the observed counts $\mathbf{Y}_i = (Y_{i,1}, \dots, Y_{i,p})^\top$ are related to the latent absolute counts via the normalization

$$\frac{Y_{i,j}}{m_i} = X_{i,j} = \frac{W_{i,j}}{\sum_{k=1}^p W_{i,k}}, \quad j = 1, \dots, p, \quad (1)$$

where $m_i = \sum_{k=1}^p Y_{i,k}$ is the subject-specific total read count and $X_{i,j}$ reflects the proportion of the j th taxon in the i th sample. Importantly, the total number of reads m_i is an artifact of sample processing and carries no information about the total microbial load in the sample. In other words, microbiome data are compositional, and spurious correlations may arise simply due to the compositionality constraint [65].

As a toy example, we generated 100 independent replicates of latent counts \mathbf{W}_i from a community with 3 bacterial taxa. The absolute abundance of each taxon is sampled independently from a negative binomial distribution with a mean of 150 and dispersion parameter of 0.25. Although the taxa are independent, naive application of Pearson correlations on the relative abundances X_i would suggest significant negative associations between some taxa, as shown in Fig. 1.

Several recent methods address this challenge of spurious correlations by correcting the associations induced from the compositional effects. Below we describe in details four such methods.

2.1.1 ReBoot

One of the most commonly used methods for microbial network inference is ReBoot [25]. This method combines permutation–renormalization and bootstrap to assess the significance of pairwise associations for compositional data. For a user-specified (dis)similarity measure (e.g. Pearson/Spearman correlation, Bray–Curtis and Kullback–Leibler dissimilarities), ReBoot uses permutation–renormalization to construct a null distribution of expected association due solely to compositionality. Specifically, for each pair of taxa, ReBoot permutes the relative abundance of one taxon, renormalizes the abundance of each sample by its total, and computes the (dis)similarity between the two taxa. ReBoot uses bootstrap to construct the

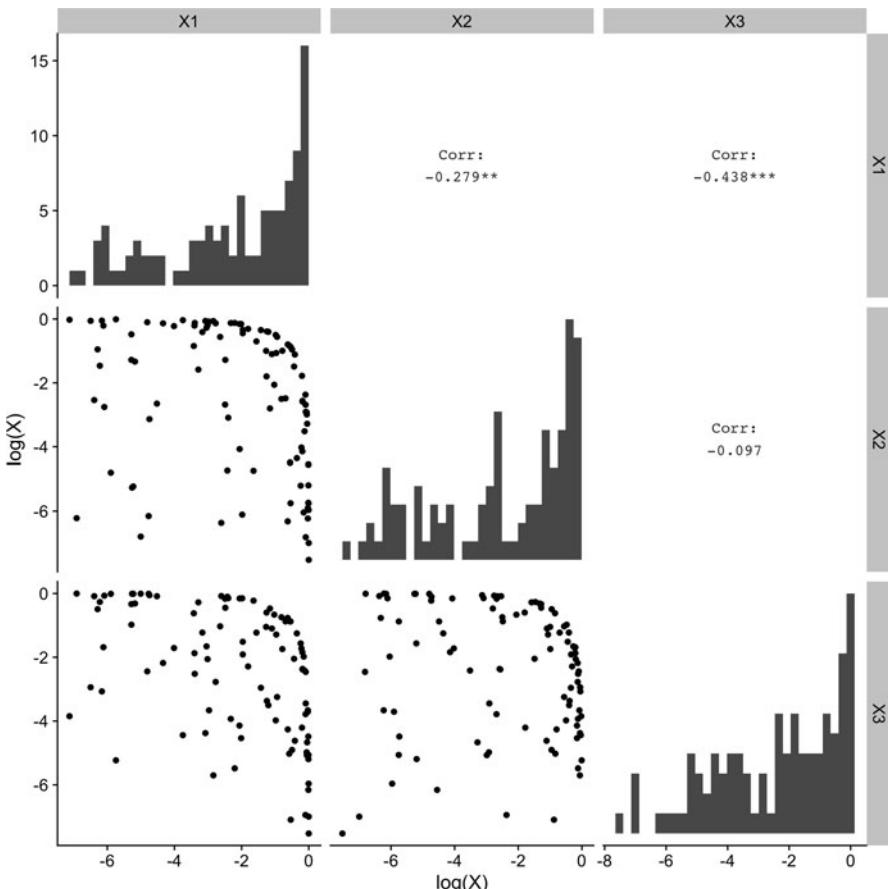


Fig. 1 Illustration of spurious correlations between log-transformed relative abundances of 3 independent taxa. The marginal distribution of each taxon is shown along the diagonal. The upper panels show the Pearson correlation between each pair of taxa, where stars indicate that the p -value of the correlation test is less than 0.01 (**) or 0.001 (***) . The lower panels show the scatter plots of log-transformed relative abundances. A pseudocount of 1 was added to all counts to prevent from taking logarithm of zeros

alternative distribution, which reflects the confidence interval around observed association. To assess the significance of each pairwise association, ReBoot tests whether the two distributions have the same mean using a z -test with pooled variance. The method requires specifying the association measure and the number of bootstrap subsamples. It is also possible to build a consensus network using multiple association measures [25, 81].

ReBoot is implemented in the Bioconductor package `ccrepe` [71]. This function takes as input proportion data, which can be obtained by the total sum normalization. By default, ReBoot uses the Spearman correlation as the similarity

measure among taxa and 1000 bootstrap subsamples to evaluate the significance for each entry of the correlation network.

2.1.2 SparCC

SparCC [31] aims to estimate the latent correlation $\rho^0 = (\rho_{j,k}^0)_{p \times p} = \text{Cor}(\log \mathbf{W}_i)$ by exploiting the connection between $\rho_{j,k}^0$ and the Aitchison variation [1]

$$T_{j,k} = \text{Var} \left(\log \frac{X_{i,j}}{X_{i,k}} \right).$$

By scale invariance,

$$T_{j,k} = \sigma_{j,j}^0 + \sigma_{k,k}^0 - 2\rho_{j,k}^0 \sqrt{\sigma_{j,j}^0 \sigma_{k,k}^0}. \quad (2)$$

Exact solution to (2) is infeasible without additional assumptions because the latent variances are unknown. The SparCC algorithm provides an approximate solution by assuming that the average correlation between each taxon and other taxa is small. This assumption allows one to solve for the approximated variance $\sigma_{j,j}^0$ using p equations

$$\sum_{k=1}^p T_{j,k} \approx (p-1)\sigma_{j,j}^0 + \sum_{k:k \neq j} \sigma_{k,k}^0, \quad j = 1, \dots, p. \quad (3)$$

Given estimates of $\sigma_{j,j}^0$ for $j = 1, \dots, p$, the correlation $\rho_{j,k}^0$ follows immediately from Eq. (2).

When calculating the Aitchison variation $T_{j,k}$, SparCC takes advantage of the observed counts \mathbf{Y}_i to obtain robust estimates of taxa correlations. Assuming that the counts \mathbf{Y}_i are generated by multinomial sampling of the proportions \mathbf{X}_i and a uniform prior, SparCC obtains the posterior distribution of \mathbf{X}_i given observed \mathbf{Y}_i . The algorithm then repeatedly draws proportions from this posterior distribution and estimates the taxa correlations, generating a distribution of each pairwise correlation. SparCC takes the median value of each correlation distribution as the final estimate. To handle scenarios where the assumption of small average correlation is violated, Friedman et al. [31] also introduced the iterative SparCC algorithm, which uses an iterative procedure to remove pairs of taxa whose correlation exceeds a given threshold. Finally, SparCC evaluates the significance of the inferred correlations using a bootstrap procedure.

2.1.3 CCLasso

Because SparCC only provides an approximate solution to (2), its correlation estimate is not guaranteed to be positive definite. To address this limitation, Fang et al. [22] proposed CCLasso to estimate all entries in the latent covariance Σ^0 simultaneously. For a compositional vector X_i with $X_{i,j} > 0$, let

$$\mathbf{Z}_i = \left(\log \frac{X_{i,1}}{g(X_i)}, \dots, \log \frac{X_{i,p}}{g(X_i)} \right)^\top$$

denote its centered log-ratio (clr) transformation, where $g(X)$ stands for the geometric mean of X . Let $F = I_p - \mathbf{1}_p \mathbf{1}_p^\top / p$, where I_p is the p -dimensional identity matrix and $\mathbf{1}_p$ is a p -dimensional vector of ones. Let $\Sigma_X = \text{Var}(\log X_i)$ denote the population covariance of $\log X_i$ and $\hat{\Sigma}_X$ be its sample version. By scale invariance, $\mathbf{Z}_i = F \log X_i = F \log W_i$. It follows immediately that

$$F \Sigma_X F = F \Sigma^0 F, \quad (4)$$

because F is symmetric. Assuming that Σ^0 is sparse, CCLasso leverages Eq. (4) and proposes the ℓ_1 penalized estimator

$$\arg \min_{\Sigma > 0} \left[\frac{1}{2} \text{tr} \left\{ F(\Sigma - \hat{\Sigma}_X) F V F (\Sigma - \hat{\Sigma}_X) F \right\} + \lambda \sum_{j \neq k} |\sigma_{j,k}| \right]. \quad (5)$$

Here, $\text{tr}(\cdot)$ denotes the matrix trace operator, $V = \{\text{diag}(F \hat{\Sigma}_X F)\}^{-1}$, $\lambda > 0$ is a regularization parameter that promotes sparsity in the solution, and $\Sigma > 0$ indicates that Σ is positive definite. The optimization in (5) can be solved by the alternating direction method of multipliers (ADMM) algorithm [7]; the tuning parameter λ can be selected via cross-validation.

The sparsity assumption underlying CCLasso is stronger than that used in SparCC, which assumes small average correlation between each taxon and its neighbors. Nevertheless, the solution from CCLasso is guaranteed to be a well-defined correlation matrix. Another key difference between SparCC and CCLasso is that the latter takes as input positive proportion data. Positive taxa proportions can be obtained by adding a pseudocount of 1 to all count values and applying total sum normalization.

2.1.4 COAT

COAT [9] is another method that learns the latent covariance Σ^0 assuming that it meets certain sparsity assumptions. Unlike CCLasso, COAT leverages the fact that

the covariance of the clr-transformed vector $\Gamma^0 = \text{Var}(\mathbf{Z}_i)$ is related to Σ^0 via the relationship

$$\Gamma^0 = F \Sigma^0 F^\top. \quad (6)$$

It is then shown that the maximum entrywise difference between Σ^0 and Γ^0 is asymptotically negligible for large and sparse covariance matrices. Therefore, for large p , sparse Σ^0 can be approximately identified by Γ^0 .

This observation motivates the COAT estimator [9] defined as follows. Let $\hat{\Gamma} = (\hat{\gamma}_{j,k})$ denote the sample covariance of the clr-transformed matrix \mathbf{Z}_i . Compute the variance estimate $\hat{\theta}_{j,k} = n^{-1} \sum_{i=1}^n (Z_{i,j,k} - \hat{\gamma}_{j,k})$, where $Z_{i,j,k} = (Z_{i,j} - \bar{Z}_j)(Z_{i,k} - \bar{Z}_k)$ and \bar{Z}_j denotes the sample mean of the j th taxon. Let $S_\lambda(\cdot)$ be a general thresholding function that satisfies (a) $S_\lambda(z) = 0$ for $|z| \leq \lambda$ and (b) $|S_\lambda(z) - z| \leq \lambda$ for all $z \in \mathbb{R}$. The COAT estimator of the target covariance matrix is then $\hat{\sigma}_{j,k} = S_{\lambda_{j,k}}(\hat{\gamma}_{j,k})$ with $\lambda_{j,k} = \delta \sqrt{\hat{\theta}_{j,k}}$. The tuning parameter $\delta > 0$ is selected via cross-validation.

2.2 Learning Networks from Conditional Associations

A major limitation of marginal association measures is that they cannot distinguish direct interactions from indirect ones [17, 73]. Two taxa that are indirectly connected in the network may still be correlated. Conditional associations, such as partial correlations [2], measure the direct interactions between two taxa after removing the indirect effects from other taxa in the community. As a result, while networks estimated based on marginal association measures are used in practice, they may include false positive edges. A simple remedy is to use the correlation estimates obtained from methods based on marginal association to infer conditional associations, using, for example, the graphical lasso algorithm [32]. However, as noted earlier, not all methods yield a valid correlation estimate. In this section, we describe three methods that directly estimate conditional associations from compositional microbiome data.

2.2.1 SPIEC-EASI

A popular approach of modeling the unknown absolute counts W_i is to use a log-normal distribution. This continuous approximation to the latent count data works well in a variety of settings [58, 64]. Under the log-normal model, the inverse covariance matrix $\Omega^0 = (\omega_{j,k}^0)_{p \times p} = (\Sigma^0)^{-1}$ defines a conditional association graph among the taxa; that is, there is an edge between taxa j and k if and only if $\omega_{j,k}^0 \neq 0$. Therefore, SPIEC-EASI [46] aims to estimate a

partial correlation network defined by the inverse covariance Ω^0 . Because the absolute abundances are unknown, SPIEC-EASI exploits the relationship in (6) and estimates Ω^0 approximately by $(\Gamma^0)^{-1}$ from the clr-transformed data Z_i . In high-dimensional settings, where the number of variables exceeds the sample size, the sample covariance $\hat{\Gamma}$ is not invertible. To address this challenge, SPIEC-EASI solves for a sparse Ω^0 using neighborhood selection [61] or the penalized maximum likelihood estimation [32, 69]. Both approaches can asymptotically recover the partial correlation network structure provided that the true network in Ω^0 satisfies certain assumptions [20]. We use the latter approach by solving the following optimization problem:

$$\hat{\Omega} = \arg \min_{\Omega > 0} \left\{ \text{tr}(\Omega \hat{\Gamma}) - \log \det(\Omega) + \lambda \sum_{j \neq k} |\omega_{j,k}| \right\}, \quad (7)$$

where $\lambda > 0$ is a regularization parameter that controls the sparsity in Ω . We choose the tuning parameter λ based on stability selection [52, 96].

It is worth noting that the performance of SPIEC-EASI (and COAT) depends on how well Σ^0 can be approximated by Γ^0 . However, Eq. (6) implies that the approximation error $\text{tr}(\Sigma^0 - \Gamma^0)$ is determined by the spectral properties of the covariance Σ^0 . When taxa are highly correlated and/or when there is heteroscedasticity among the latent abundances [43], Σ^0 can have a large condition number, in which case SPIEC-EASI (and COAT) may perform poorly [23].

2.2.2 gCoda

Unlike SPIEC-EASI that only approximately estimates the inverse covariance matrix, gCoda estimates Ω^0 directly from the relative abundances [23]. Denote by $M_i = \sum_{j=1}^p W_{i,j}$ the unknown total absolute abundance. By definition, $\log X_{i,j} = \log W_{i,j} - \log M_i$ for $j = 1, \dots, p$. Assuming that the unknown absolute abundances $\log W_i$ follow a log-normal distribution with mean μ and inverse covariance Ω^0 , the joint distribution function of $(X_i, \log M_i)$ is

$$f(X_i, \log M_i) = (2\pi)^{-\frac{p}{2}} |\Omega^0|^{\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\log W_i - \mu)^\top \Omega^0 (\log W_i - \mu) \right\},$$

where $\log W_i = \log X_i + \mathbf{1}_p^\top \log M_i$. It is straightforward to derive the conditional distribution of $\log M_i$ given X_i because $\log M_i = (\mathbf{1}_p^\top \Omega^0 \mathbf{1}_p)^{-1} \mathbf{1}_p^\top \Omega^0 (\log W_i - \log X_i)$. Therefore, we have the marginal likelihood function of X_i as

$$f(X_i) = (2\pi)^{-\frac{p-1}{2}} \left(\frac{|\Omega^0|}{\mathbf{1}_p^\top \Omega^0 \mathbf{1}_p} \right)^{\frac{1}{2}} \exp \left\{ -\frac{1}{2} (\log X_i - \mu)^\top Q (\log X_i - \mu) \right\}, \quad (8)$$

where

$$Q = \Omega^0 - \frac{\Omega^0 \mathbf{1}_p \mathbf{1}_p^\top \Omega^0}{\mathbf{1}_p^\top \Omega^0 \mathbf{1}_p}.$$

The covariance term $(\log X_i - \mu)(\log X_i - \mu)^\top$ in (8) depends on the unknown mean parameter μ and is estimated by the sample covariance $\hat{\Sigma}$ of $\log X_i$. For a tuning parameter $\lambda > 0$, gCoda solves for a sparse Ω^0 by minimizing the penalized negative log-likelihood of observed relative abundances:

$$\arg \min_{\Omega > 0} \left\{ \text{tr}(\hat{\Sigma} Q) - \log \det(\Omega) + \log(\mathbf{1}_p^\top \Omega \mathbf{1}_p) + \lambda \sum_{j \neq k} |\omega_{j,k}| \right\}. \quad (9)$$

Compared to (7), the optimization problem in (9) has an extra term $\log(\mathbf{1}_p^\top \Omega \mathbf{1}_p)$ and is solved by an efficient Majorization–Minimization algorithm [39]. The tuning parameter λ is selected via the extended Bayesian information criterion [10].

2.2.3 SPRING

Both SPIEC-EASI and gCoda assume that the unknown absolute abundances are well approximated by a log-normal distribution. In practice, microbial count data are often sparse with an excess of zeros, skewed, and sometimes have multiple modes [81]. Zeros are commonly dealt with by adding a small pseudocount to all count values [9, 22, 23, 31, 46], which may unfairly bias rare taxa [81]. To deal with high sparsity and non-normality, SPRING [89] uses a modified clr transformation of zero-inflated count data. Without loss of generality, let $X_i = (X_{i,1}, \dots, X_{i,q_i}, 0, \dots, 0)^\top$ denote the observed relative abundances in the i th sample, where only the first $q_i \leq p$ entries are strictly positive. The modified clr transformation is defined as follows:

$$\text{mclr}_\epsilon(X_i) = \left(\log \frac{X_{i,1}}{g(X_{i,1:q_i})} + \epsilon, \dots, \log \frac{X_{i,q_i}}{g(X_{i,1:q_i})} + \epsilon, 0, \dots, 0 \right)^\top,$$

with the adjustment

$$\epsilon = \left| \min_{i,j: X_{i,j} > 0} \log \frac{X_{i,j}}{g(X_{i,1:q_i})} \right| + c,$$

for a small constant $c > 0$. The idea is to apply the clr transformation only to positive proportions and shift all transformed values to be strictly positive. By doing so, mclr preserves the original ordering of entries in the composition X_i while avoiding the use of arbitrary pseudocounts.

The mclr-transformed abundances are zero-inflated and hence can be modeled via the truncated Gaussian copula model. The correlation of this copula model

serves as a surrogate for the desired correlation Σ^0 . SPRING uses a rank-based estimator for Σ^0 [90] and estimates the partial correlation network using the graphical lasso in (7). As done in SPIEC-EASI, the tuning parameter is chosen via stability selection [52, 96].

3 Data-Generating Models

To reflect the complexity of microbial abundance data, we generated benchmark data sets from the American Gut Project [60] and synthetic data sets from parametric distributions. These data sets cover a wide range of settings with various sample sizes and numbers of taxa.

3.1 Null Models

We use two null models to assess how various methods handle spurious (partial) correlations induced by the compositional constraint.

In the first model, we generated benchmark data sets using the American Gut Project data available in the R package `SpiecEasi`. This data set consists of the count abundances of 127 taxa from 289 observations. The total number of reads per observation ranges from 12,007 to 42,331, and the percentage of zeros is about 31%. We first processed this data set to ensure even depth across observations. To this end, we rarefied this data set such that the total number of reads is 1000 for all observations, using the “`rarefy_even_depth`” function in the `phyloseq` R package. Rarefaction also increased the percentage of zeros in the data to be about 60%, which allows evaluation of all methods under high sparsity scenarios. For each taxon, we fit a zero-inflated negative binomial (ZINB) model to its rarefied counts and generated new counts under the fitted model for various sample sizes. The ZINB model is justified because it adequately describes the over-dispersion of observed taxa counts [46]. Importantly, we sampled the abundance of each taxon independently so as to create a null setting with no correlations. The abundance table generated in this fashion has roughly equal number of reads across observations. To reflect the large variation in sequencing depth often seen in practice [78], in the last step, we multiplied each observation by an integer uniformly sampled between 1 and 10. The final abundance table is of size $p \times n$, where $p = 127$ and $n \in \{100, 200, 300, 500\}$.

In the second model, the abundance table was generated from the Dirichlet-multinomial (DM) distribution, which is commonly used to model microbial count data [11, 37]. The Dirichlet distribution imposes compositional constraint on observed counts but still defines a null model because it is equivalent to normalized independent Gamma processes. For $p = 200$ and $n \in \{100, 200, 300, 500\}$, we first

sampled absolute abundances $W_{i,j}$ from a log-normal distribution with mean μ_j and standard deviation 1.5, where $\mu_j \sim \text{Unif}[0, 4]$. We then computed the compositions

$$\phi_{i,j} = \frac{\exp(W_{i,j})}{\sum_{k=1}^p \exp(W_{i,k})}$$

and generated count data from the DM distribution $\text{DM}(m_i, \alpha\phi_{i,1}, \dots, \alpha\phi_{i,p})$. Here, $\alpha = 100$, and the depth m_i was sampled uniformly between m_0 and $10m_0$, where $m_0 = 12,007$ is the minimum sequencing depth in the American Gut Project data set.

3.2 Copula Models

The copula model [46] allows one to generate taxa counts with a pre-specified correlation matrix and marginal distributions. We used this model to assess the sensitivity and specificity of various network estimation methods. To this end, we first constructed a $p \times p$ precision matrix Ω^0 , whose structure was generated as an Erdős-Rényi random graph with $3p$ edges. Off-diagonal entries of Ω^0 were uniformly sampled from $[-3, -2] \cup [2, 3]$. The positive definiteness of Ω^0 was guaranteed by setting the diagonal entries to be

$$|\Lambda_{\min}(\Omega^0 + I_p)| + c, \quad (10)$$

where $\Lambda_{\min}(A)$ denotes the smallest eigenvalue of the matrix A and the constant c determines the condition number of Ω^0 . The correlation matrix Σ^0 is then given by

$$\Sigma_{j,k}^0 = \frac{(\Omega^0)_{j,k}^{-1}}{\sqrt{(\Omega^0)_{j,j}^{-1}(\Omega^0)_{k,k}^{-1}}}.$$

Given the correlation, we used the rarefied taxa counts and the ZINB model described in Sect. 3.1 to generate the marginal counts for each taxon. We generated $p = 127$ taxa with varying sample size $n \in \{100, 200, 300, 500\}$. To reflect the variation in sequencing depth, we also scaled the counts in each observation as described in Sect. 3.1.

Note in practice we do not know the (partial) correlations among taxa, nor do we know how well conditioned the (inverse) covariance matrix is. When the number of features p is large, the empirical covariance matrix is known to be poorly conditioned [82]. For example, the empirical covariance of the American Gut data after clr transformation has a condition number 10^{18} , because its smallest eigenvalue is very close to zero. To avoid generating poorly conditioned Ω^0 , we set the constant c in (10) such that the resulting Ω^0 has a condition number 1000, which is similar to the ratio between the largest and the second smallest eigenvalues of the empirical covariance.

3.3 Logistic-Normal Model

In addition to the copula model, we also generated continuous relative abundance data from the logistic-normal distribution for a pre-specified partial correlation matrix. For $p = 200$ and $n \in \{100, 200, 300, 500\}$, we sampled the absolute abundances W_i from a log-normal distribution with mean vector μ and covariance matrix Σ^0 . The mean parameter is $\mu_j \sim \text{Uniform}(0, 4)$, $j = 1, \dots, p$, and the covariance matrix Σ^0 was generated following the procedure described in Sect. 3.2. The observed relative abundances X_i are defined as

$$X_{i,j} = \frac{\exp(W_{i,j})}{\sum_{k=1}^p \exp(W_{i,k})}, \quad j = 1, \dots, p,$$

for all $i = 1, \dots, n$.

4 Results

Under the null models, all taxa are independent in both marginal and conditional associations. Because most methods only provide estimates of (partial) correlations, we evaluated whether they detect any spurious (partial) correlations by calculating the false positive rate

$$\frac{\text{FP}}{p(p-1)/2}, \quad (11)$$

where FP stands for the number of false positives. For CCLasso and COAT, FP is calculated by the number of pairs (j, k) for which $|\hat{\Sigma}_{j,k}| > 0$. For SPIEC-EASI, gCoda, and SPRING, we determined FP by counting the number of pairs for which $|\hat{\Omega}_{j,k}| > 0$. ReBoot and SparCC conduct inference and return the p -values for testing the null $H_0 : \Sigma_{j,k}^0 = 0$ for all $1 \leq j < k \leq p$. Thus, FP is determined by the total number of p -values that are less than 0.05.

Under the alternative models, we compared the performance of all methods except ReBoot in estimating the partial correlation network. While acknowledging the fact that SparCC, CCLasso, and COAT are not optimized to estimate partial correlations, to provide a fair comparison, we evaluate all methods based on the same target. To this end, we augmented SparCC, CCLasso, and COAT with an additional step to estimate a partial correlation network. Specifically, given an optimal correlation estimate $\hat{\Gamma}$, the partial correlation network can be solved by the graphical lasso in (7). ReBoot was excluded because it is inference-based and does not provide an estimate of the correlation or partial correlation matrix. We used the receiver operating characteristic (ROC) curve [30] to assess each method's sensitivity and specificity in estimating the true network. The ROC curve is a plot of

the false positive rate (FPR) against the true positive rate (TPR). Let FPR and TPR be defined, respectively, as

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

where FP, TP, FN, and TN refer to the number of false positives, true positives, false negatives, and true negatives, respectively. Given the true network Ω^0 and an estimate $\hat{\Omega}$, we declare a true positive between taxa j and k if $|\hat{\Omega}_{j,k}| > 0$ and $|\Omega_{j,k}^0| > 0$.

4.1 Spurious (Partial) Correlations

Figure 2 shows the average false positive rates of various methods from 100 replications under the two null models. The false positive rates of estimation-based methods (COAT, CCLasso, SPIEC-EASI, gCoda, and SPRING) are in general smaller than inference-based methods (ReBoot and SparCC), which can be attributed to the small threshold used in calling a positive discovery. Still, SPRING and gCoda occasionally yield larger false positive rates than COAT, CCLasso, and SPIEC-EASI. Between inference-based methods, SparCC yields slightly larger false

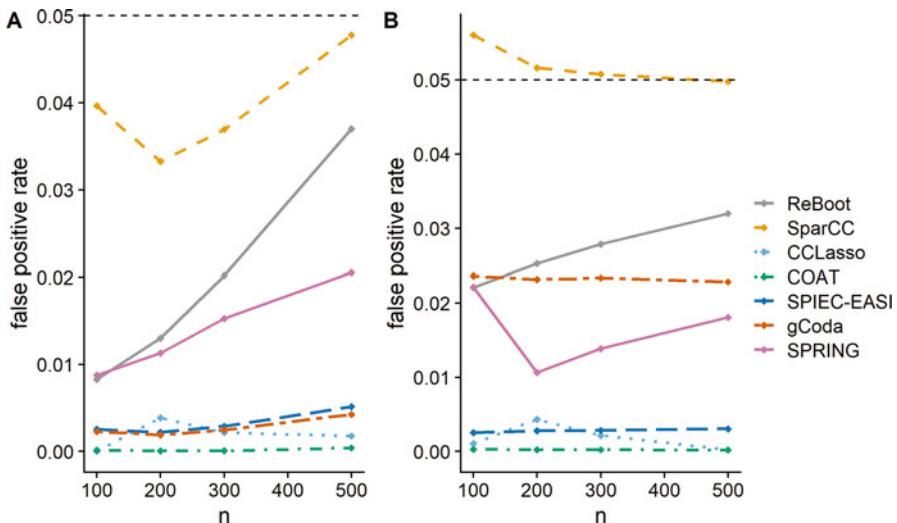


Fig. 2 Proportion of spurious (partial) correlations for various methods under null model 1 (**a**) and null model 2 (**b**). The x -axis shows the sample size and y -axis indicates the false positive rate, which defines the proportion of spurious (partial) correlations

positive rates than ReBoot, but both control the false positive rate at the nominal significance level.

4.2 Performance in Network Discovery

Figure 3 shows the ROC curves for different methods under the copula model. All methods except CCLasso perform well in recovering the true partial correlation network. Importantly, SPRING shows superior performance, especially when the sample size is large, because it explicitly accounts for the high sparsity in the data, which is around 60%. Depending on the taxonomic level of the analysis, microbiome data can have as many as 90% zeros [64]. We anticipate that methods such as SPRING work well with very sparse data sets, both by accounting for the large number of zeros and by relaxing the parametric model assumptions. SparCC works reasonably well, because most entries in the correlation network are small. In addition, SparCC samples taxa proportions from its posterior distribution given observed counts, which improves robustness against sparsity in data. This is in contrast to most methods, including CCLasso, COAT, SPIEC-EASI and gCoda, which apply total sum normalization or clr transformation to the observed counts

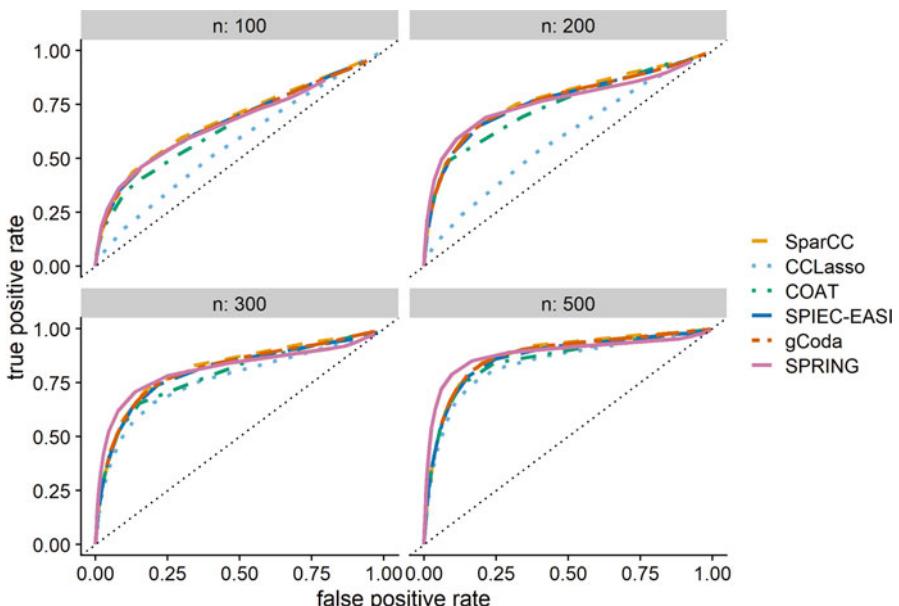


Fig. 3 Average ROC curves for various methods over 20 replications under the copula model. All methods except CCLasso show comparable performance in recovering the partial correlation network. When the sample size is small ($n = 100, 200$), CCLasso performs poorly. SPRING yields the best overall performance

without accounting for the random sampling of taxa counts. Among all methods, CCLasso performs the worst because the correlation matrix is not exactly sparse. Both COAT and SPIEC-EASI estimate the empirical correlation matrix from the clr-transformed data before passing it to the graphical lasso step in (7). However, the soft thresholding step in COAT, although desirable when estimating the correlations, may have negatively impacted its performance in estimating the partial correlation network.

Figure 4 shows the ROC curves for different methods under the logistic-normal model, where the data are positive proportions. With proportion data, SPIEC-EASI, gCoda, and SPRING generally work well, whereas SparCC, COAT, and CCLasso perform poorly. The performance of SparCC is not surprising, because it was originally proposed to estimate correlations from count data. As is the case in the copula model, the true correlation network is not sparse, which violates the model assumption underlying CCLasso. So, CCLasso performs the worst among all methods. The soft thresholding step used in COAT in estimating the empirical correlations again has negatively impacted its performance in recovering the partial correlation network. But the performance of COAT improved substantially with the larger sample size, $n = 500$.

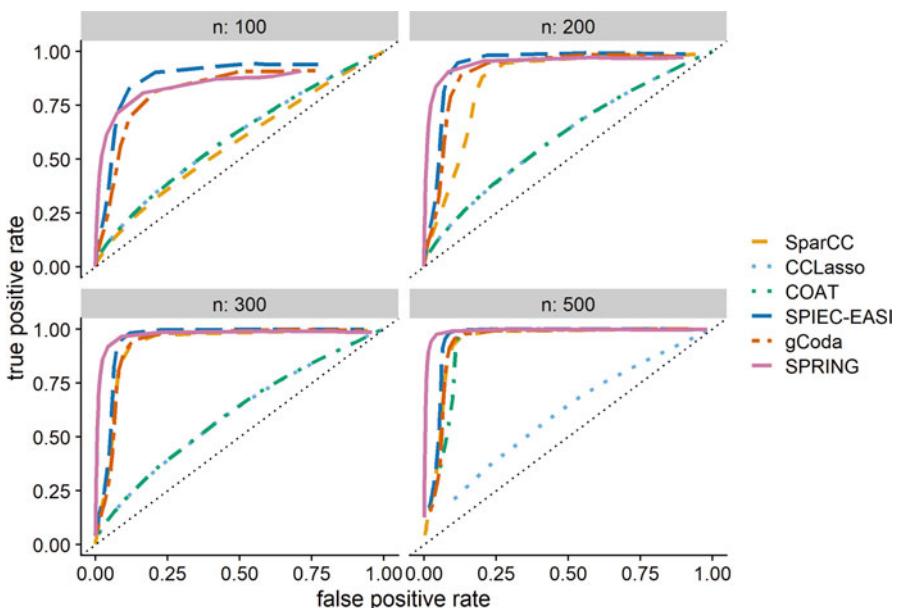


Fig. 4 Average ROC curves for different methods over 20 replications under the logistic-normal model. SPIEC-EASI, gCoda, and SPRING perform well under all sample sizes. SparCC performs poorly when the sample size is small ($n = 100$), whereas CCLasso did poorly across all sample sizes. When the sample size is large ($n = 500$), the performance of COAT improves substantially

4.3 Case Studies in R

In this section, we illustrate analysis of the American Gut Project data set using the aforementioned methods.

ReBoot, SPIEC-EASI, and SPRING are available as R/Bioconductor packages, whereas SparCC is implemented in the R package *SpiecEasi*. The other methods are only available in the form of R functions, which can be downloaded from the authors' GitHub pages (see Table 1). We begin by importing the necessary R/Bioconductor packages and functions.

```
357 library(ccrepe) # for ReBoot
358 # library(devtools)
359 # install_github("zdk123/SpiecEasi")
360 library(SpiecEasi)
361 library(pulsar)
362 # install_github("GraceYoon/SPRING")
363 library(SPRING)
364 source("cclasso.R")
365 source("coat.R")
366 source("gcoda.R")
```

We will use the reference data set available in the R package *SpiecEasi*, which consists of counts of 127 taxa over 289 subjects. Not all methods take as input the observed count data. We thus derive the proportion data by applying total sum normalization to count data, both before and after adding the pseudocount.

```
367 data("amgut1.filt")
368 reference <- amgut1.filt
369
370 ## apply total sum normalization to observed counts
371 proportions <- sweep(reference,1,STATS = rowSums(reference),FUN="/")
372
373 ## apply total sum normalization to zero corrected counts
374 if(any(reference==0)){
375   reference.adj = reference + 1
376 }
377 proportions.adj <- sweep(reference.adj,1,STATS = rowSums(
  reference.adj),FUN="/")
```

We are now ready to apply the aforementioned methods to estimate the correlation and partial correlation network from the reference data set. The following code shows how to infer the correlation network:

```
378 ## --- ReBoot ---
379 reboot.res <- ccrepe(x = proportions)
380 reboot.qval.matrix <- reboot.res$q.values
381 diag(reboot.qval.matrix) <- 1
382 reboot.network <- adj2igraph((reboot.qval.matrix<0.01)) #
  construct network by thresholding the q values
383
384 ## --- SparCC ---
```

```

385 sparcc.res = SpiecEasi::sparccboot(data = reference, R=1000,
386   sparcc.params = list(iter = 20, inner_iter = 10, th = 0.1)) # 
387   This takes a long time!!!
388 sparcc.pval.matrix <- diag(1, ncol(reference))
389 sparcc.pval.matrix[upper.tri(sparcc.pval.matrix)] <- pval.
390   sparccboot(sparcc.res)$pvals
391 sparcc.pval.matrix <- sparcc.pval.matrix + t(sparcc.pval.matrix)
392 sparcc.network <- adj2igraph((sparcc.pval.matrix<0.05)))
393
394 ## --- CClasso ---
395 cclasso.res <- cclasso(x = reference, counts = T, pseudo = 1, k_
396   cv = 3, lam_int = c(1e-4, 3), k_max = 20, n_boot = 1000) #
397   both k_cv and k_max are arguments necessary for tuning
398   parameter selection
399 cclasso.network = cclasso.res$cor_w # extract the correlation
400   network
401
402 ## --- COAT ---
403 coat.res = coat(proportions.adj, soft=1)
404 coat.network = coat.res$corr

```

After obtaining the network, one can use the `igraph` package to visualize the network (see Fig. 5). Alternatively, Fig. 6 shows a visualization of the correlations using the `GGally` package.

Similarly, we can construct the partial correlation network with SPIEC-EASI, gCoda, and SPRING as follows. SPIEC-EASI and gCoda require a count data matrix as input and performs internal pseudocount correction if needed. Both SPIEC-EASI and SPRING perform stability selection to select partial correlations that are the most frequently selected over many subsamples. It is recommended to experiment with different parameter settings so as to yield interpretable results. Figure 7 shows a visualization of the partial correlation network estimated by SPIEC-EASI.

```

398 ## --- SPIEC-EASI ---
399 spieceasi.res <- spiec.easi(reference, method="glasso", pulsar.
400   select = T, pulsar.params = list(thresh=0.05, subsample.ratio
401   =0.8, rep.num = 20), lambda.min.ratio=0.01, nlambda=50) #
402   pulsar parameters can be customized
403 spieceasi.network <- adj2igraph(getOptNet(spieceasi.res))
404
405 ## --- gCoda ---
406 gcoda.res <- gcoda(reference, counts = T, pseudo = 1, lambda.min.
407   ratio=1e-3, nlambda=50) # specify pseudocount value if the
408   input is count data
409 gcoda.network <- gcoda.res$opt.icov
410
411 ## --- SPRING ---
412 spring.res <- SPRING(proportions, quantitative = F, subsample.
413   ratio = 0.8, rep.num = 20) # if input is compositional, set
414   quantitative to be FALSE
415 opt.K <- spring.res$output$stars$opt.index

```



Fig. 5 The correlation network inferred by ReBoot. Each node represents a taxon, and each edge represents a significant correlation after controlling the false discovery rate at 1%

```

409 sp.network.binary = as.matrix(spring.res$fit$est$path[[opt.K]]) # 
  a binary network indicating presence/absence of edges
410 sp.network.weighted <- as.matrix(SpiecEasi::symBeta(spring.res$ 
  output$est$beta[[opt.K]], mode = "maxabs")) # a weighted
  network matrix

```

From a computation perspective, COAT, CCLasso, and gCoda take less than 10 s to analyze the American Gut data set available in the R package SpiecEasi. The computation time for ReBoot, SPIEC-EASI, and SPRING ranges from 2 to 5 min. SparCC is the slowest, requiring about an hour to finish the 1000 bootstrap subsampling. SpiecEasi and SPRING would also require significantly more computational time if their tuning parameters were chosen via stability selection.

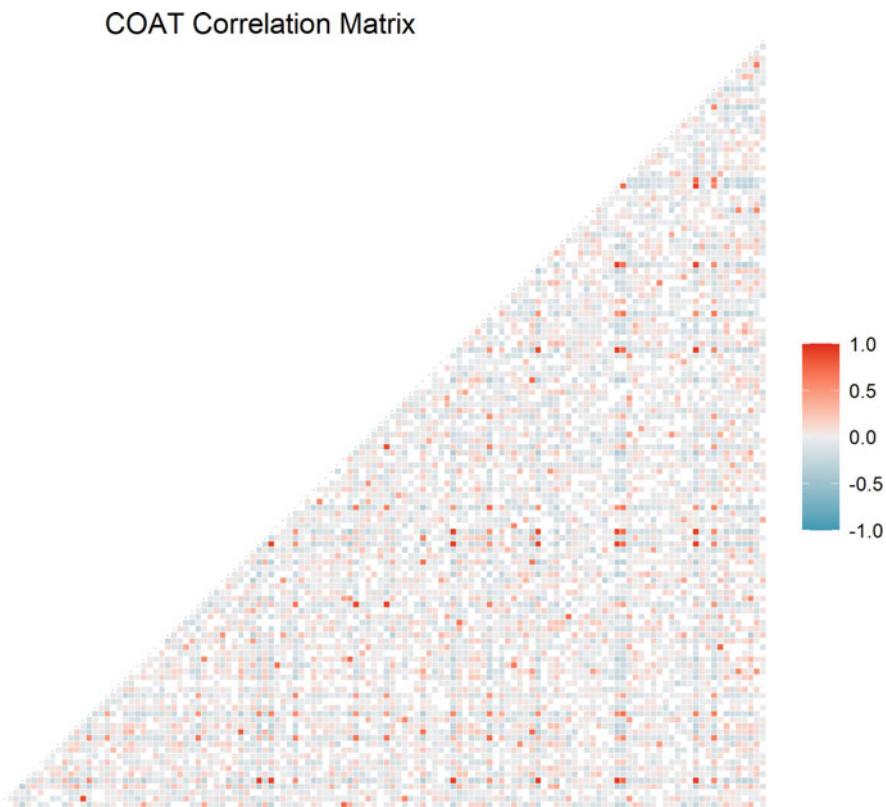


Fig. 6 The correlations estimated by COAT

5 Future Directions

Networks are increasingly used to glean insight into interaction mechanisms among microbes [24, 26, 67]. Due to their dynamic nature and our limited knowledge of microbial interactions, this is often achieved by learning microbial interaction networks from (relative) abundance data. The methods reviewed in this chapter aim to address the challenges of inferring microbial interaction networks and provide reliable estimates of network structures. These methods impose different assumptions on data distribution. As our numerical analysis indicates, they perform reasonably well when these assumptions are satisfied and may fail otherwise. An important area of future research is to develop more flexible estimation frameworks for microbial interaction networks similar to those developed for other types of omics data [50, 87, 91] as well as semi- and non-parametric methods for inferring graphical models [19, 28, 48, 51, 53, 80, 85].

Measures of uncertainty for network estimates, such as confidence intervals and p -values, are critical for reproducible and generalizable scientific discovery. This

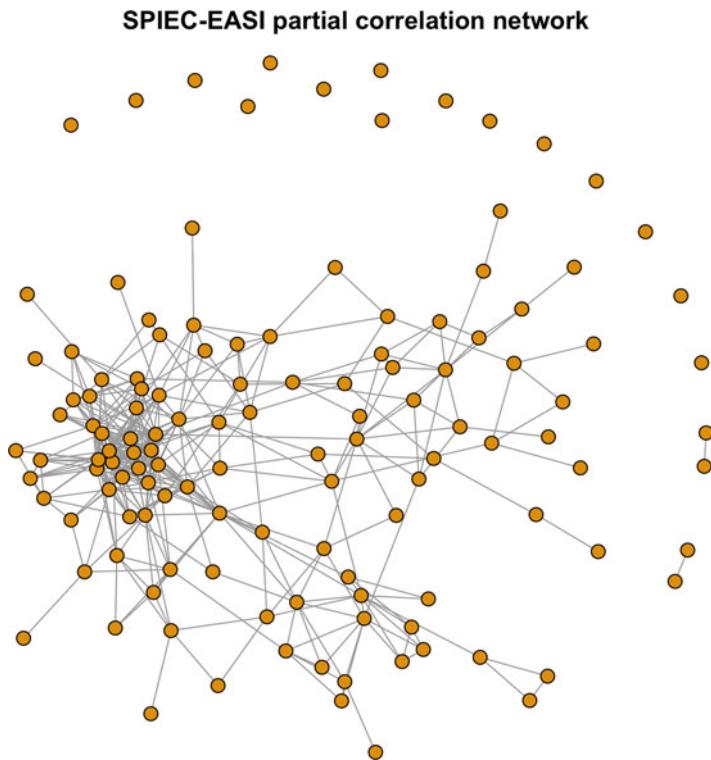


Fig. 7 The partial correlation network estimated by SPIEC-EASI. Each node represents a taxon. Edges represent those that pass the stability selection threshold

is especially important since network estimates have high level of uncertainty [72]. Fortunately, significant progress has been made over the past couple of years in inference for graphical models [41, 42, 68, 83, 92]. Although inference procedures for marginal associations among microbes have been developed [25, 31], this remains a fruitful area of future research.

Despite its importance and considerable recent attention, network estimation is often the first step in analysis pipelines that aim to discover underlying disease mechanisms. A powerful tool for achieving this goal is to identify differences among networks from different conditions, e.g. microbial networks of patients compared to healthy individuals. This approach, which is known as *differential network analysis* [40, 73], has been the focus of considerable research in graphical modeling [5, 16, 35, 45, 55, 66, 70, 84, 93, 95, 97–99], including approaches for microbial interaction networks [8, 36]; see [73] for a more comprehensive review. Given the potential challenges of inferring changes in network structures based on quantitative measures [98], a fruitful area of future research is investigating methods for identifying changes in microbiome networks based on qualitative hypothesis tests [38]. Future work may also focus on interactions among microbes and other

omics measures [12, 14, 27, 43, 88], as well as network-based pathway enrichment analysis [56]. Discovering causal effects of microbes on each other is also critical for developing effective treatments [29, 47]. However, despite significant progress in causal discovery from observational data [13, 20, 33, 44, 63, 75], further progress in this area requires effective validation experiments [6, 76], which remain challenging in microbiome settings [54].

Acknowledgments This work was partially funded by grants from the National Science Foundation (DMS-1561814) and the National Institutes of Health (R01 GM114029, R01 GM129512, and R01 GM133848).

References

1. Aitchison, J.W.: *The Statistical Analysis of Compositional Data*. Chapman & Hall, Ltd., Boca Raton (1986)
2. Baba, K., Shibata, R., Sibuya, M.: Partial correlation and conditional correlation as measures of conditional independence. *Austr. New Zeal. J. Stat.* **46**(4), 657–664 (2004)
3. Barabasi, A.L., Oltvai, Z.N.: Network biology: understanding the cell’s functional organization. *Nat. Rev. Genet.* **5**(2), 101–113 (2004)
4. Barabási, A.L., Gulbahce, N., Loscalzo, J.: Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.* **12**(1), 56–68 (2011)
5. Belilovsky, E., Varoquaux, G., Blaschko, M.B.: Testing for differences in Gaussian graphical models: applications to brain connectivity. In: Lee, D.D., Sugiyama, M., Luxberg, U.V., Guyon, I., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, vol. 29, pp. 595–603. Curran Associates, Inc., Red Hook, NY (2016)
6. Blainey, P.C.: The future is now: single-cell genomics of bacteria and archaea. *FEMS Microbiol. Rev.* **37**(3), 407–427 (2013)
7. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **3**(1), 1–122 (2010)
8. Cai, T., Li, H., Ma, J., Xia, Y.: Differential Markov random field analysis with an application to detecting differential microbial community networks. *Biometrika* **106**(2), 401–416 (2019)
9. Cao, Y., Lin, W., Li, H.: Large covariance estimation for compositional data via composition-adjusted thresholding. *J. Am. Stat. Assoc.* **114**(526), 759–772 (2019)
10. Chen, J., Chen, Z.: Extended Bayesian information criteria for model selection with large model spaces. *Biometrika* **95**(3), 759–771 (2008)
11. Chen, J., Li, H.: Variable selection for sparse Dirichlet-multinomial regression with an application to microbiome data analysis. *Ann. Appl. Stat.* **7**(1), 418–442 (2013)
12. Chen, S., Witten, D., Shojaie, A.: Selection and estimation for mixed graphical models. *Biometrika* **102**(1), 47–64 (2015)
13. Chen, W., Drton, M., Wang, Y.S.: On causal discovery with an equal-variance assumption. *Biometrika* **106**(4), 973–980 (2019)
14. Cheng, J., Li, T., Levina, E., Zhu, J.: High-dimensional mixed graphical models. *J. Comput. Graph. Stat.* **26**(2), 367–378 (2017)
15. Coyte, K.Z., Schluter, J., Foster, K.R.: The ecology of the microbiome: networks, competition, and stability. *Science* **350**(6261), 663–666 (2015)
16. Danaher, P., Wang, P., Witten, D.M.: The joint graphical lasso for inverse covariance estimation across multiple classes. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **76**(2), 373–397 (2014)

17. De La Fuente, A., Bing, N., Hoeschele, I., Mendes, P.: Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics* **20**(18), 3565–3574 (2004)
18. Ding, T., Schloss, P.D.: Dynamics and associations of microbial community types across the human body. *Nature* **509**(7500), 357–360 (2014)
19. Dobra, A., Lenkoski, A.: Copula Gaussian graphical models and their application to modeling functional disability data. *Ann. Appl. Stat.* **5**(2A), 969–993 (2011)
20. Drton, M., Maathuis, M.H.: Structure learning in graphical modeling. *Ann. Rev. Stat. Appl.* **4**, 365–393 (2017)
21. Fan, J., Lv, J.: A selective overview of variable selection in high dimensional feature space. *Stat. Sin.* **20**(1), 101 (2010)
22. Fang, H., Huang, C., Zhao, H., Deng, M.: CCLasso: correlation inference for compositional data through lasso. *Bioinformatics* **31**(19), 3172–3180 (2015)
23. Fang, H., Huang, C., Zhao, H., Deng, M.: gCoda: conditional dependence network inference for compositional data. *J. Comput. Biol.* **24**(7), 699–708 (2017)
24. Faust, K., Raes, J.: Microbial interactions: from networks to models. *Nat. Rev. Microbiol.* **10**(8), 538–550 (2012)
25. Faust, K., Sathirapongsasuti, J.F., Izard, J., Segata, N., Gevers, D., Raes, J., Huttenhower, C.: Microbial co-occurrence relationships in the human microbiome. *PLoS Comput. Biol.* **8**(7), e1002606 (2012)
26. Faust, K., Sathirapongsasuti, J.F., Izard, J., Segata, N., Gevers, D., Raes, J., Huttenhower, C.: Microbial co-occurrence relationships in the human microbiome. *PLoS Comput. Biol.* **8**(7), e1002606 (2012)
27. Faust, K., Lima-Mendez, G., Lerat, J.S., Sathirapongsasuti, J.F., Knight, R., Huttenhower, C., Lenaerts, T., Raes, J.: Cross-biome comparison of microbial association networks. *Front. Microbiol.* **6**, 1200 (2015)
28. Fellighauer, B., Bühlmann, P., Ryffel, M., Von Rhein, M., Reinhardt, J.D.: Stable graphical model estimation with random forests for discrete, continuous, and mixed variables. *Comput. Stat. Data Anal.* **64**, 132–152 (2013)
29. Fischbach, M.A.: Microbiome: focus on causation and mechanism. *Cell* **174**(4), 785–790 (2018)
30. Flach, P.A.: The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 194–201 (2003)
31. Friedman, J., Alm, E.J.: Inferring correlation networks from genomic survey data. *PLoS Comput. Biol.* **8**(9), e1002687 (2012)
32. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**(3), 432–441 (2008)
33. Ghoshal, A., Honorio, J.: Learning identifiable Gaussian Bayesian networks in polynomial time and sample complexity. In: Advances in Neural Information Processing Systems, pp. 6457–6466 (2017)
34. 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)
35. Guo, J., Levina, E., Michailidis, G., Zhu, J.: Joint estimation of multiple graphical models. *Biometrika* **98**(1), 1–15 (2011)
36. He, S., Deng, M.: Direct interaction network and differential network inference from compositional data via lasso penalized d-trace loss. *PLoS One* **14**(7), e0207731 (2019)
37. Holmes, I., Harris, K., Quince, C.: Dirichlet multinomial mixtures: generative models for microbial metagenomics. *PLoS One* **7**(2), e30126 (2012)
38. Hudson, A., Shojai, A.: Statistical inference for qualitative interactions with applications to precision medicine and differential network analysis (2020). Preprint. arXiv:2010.08703
39. Hunter, D.R., Lange, K.: A tutorial on mm algorithms. *Am. Stat.* **58**(1), 30–37 (2004)
40. Ideker, T., Krogan, N.J.: Differential network biology. *Mol. Syst. Biol.* **8**(1), 565 (2012)
41. Jankova, J., Van De Geer, S.: Confidence intervals for high-dimensional inverse covariance estimation. *Electron. J. Stat.* **9**(1), 1205–1229 (2015)

42. Janková, J., van de Geer, S.: Honest confidence regions and optimality in high-dimensional precision matrix estimation. *Test* **26**(1), 143–162 (2017)
43. Jiang, D., Armour, C.R., Hu, C., Mei, M., Tian, C., Sharpton, T.J., Jiang, Y.: Microbiome multi-omics network analysis: statistical considerations, limitations, and opportunities. *Front. Genet.* **10**, 995 (2019)
44. Kalisch, M., Bühlmann, P.: Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *J. Mach. Learn. Res.* **8**, 613–636 (2007)
45. Kim, B., Liu, S., Kolar, M.: Two-sample inference for high-dimensional Markov networks (2019). Preprint. arXiv:1905.00466
46. Kurtz, Z.D., Müller, C.L., Miralda, E.R., Littman, D.R., Blaser, M.J., Bonneau, R.A.: Sparse and compositionally robust inference of microbial ecological networks. *PLoS Comput. Biol.* **11**(5), e1004226 (2015)
47. Layeghifard, M., Hwang, D.M., Guttman, D.S.: Disentangling interactions in the microbiome: a network perspective. *Trends Microbiol.* **25**(3), 217–228 (2017)
48. Lee, K.Y., Li, B., Zhao, H.: On an additive partial correlation operator and nonparametric estimation of graphical models. *Biometrika* **103**(3), 513–530 (2016)
49. Li, H.: Microbiome, metagenomics, and high-dimensional compositional data analysis. *Ann. Rev. Stat. Appl.* **2**, 73–94 (2015)
50. Lin, L., Drton, M., Shojaie, A.: Estimation of high-dimensional graphical models using regularized score matching. *Electron. J. Stat.* **10**(1), 806–854 (2016)
51. Liu, H., Lafferty, J., Wasserman, L.: The nonparanormal: semiparametric estimation of high dimensional undirected graphs. *J. Mach. Learn. Res.* **10**(10), 2295–2328 (2009)
52. Liu, H., Roeder, K., Wasserman, L.: Stability approach to regularization selection (stars) for high dimensional graphical models. In: Advances in Neural Information Processing Systems, pp. 1432–1440 (2010)
53. Liu, H., Han, F., Yuan, M., Lafferty, J., Wasserman, L.: High-dimensional semiparametric Gaussian copula graphical models. *Ann. Stat.* **40**(4), 2293–2326 (2012)
54. Lv, X., Zhao, K., Xue, R., Liu, Y., Xu, J., Ma, B.: Strengthening insights in microbial ecological networks from theory to applications. *mSystems* **4**(3), e00124-19 (2019)
55. Ma, J., Michailidis, G.: Joint structural estimation of multiple graphical models. *J. Mach. Learn. Res.* **17**(1), 5777–5824 (2016)
56. Ma, J., Shojaie, A., Michailidis, G.: A comparative study of topology-based pathway enrichment analysis methods. *BMC Bioinf.* **20**(1), 546 (2019)
57. Maathuis, M., Drton, M., Lauritzen, S., Wainwright, M.: Handbook of Graphical Models. CRC Press, Boca Raton (2018)
58. Magurran, A.E., Henderson, P.A.: Explaining the excess of rare species in natural species abundance distributions. *Nature* **422**(6933), 714–716 (2003)
59. Margolin, A., Nemenman, I., Bassi, K., Wiggins, C., Stolovitzky, G., Dalla Favera, R., Califano, A.: ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinf.* **7**(1) (2006)
60. McDonald, D., Hyde, E., Debelius, J.W., Morton, J.T., Gonzalez, A., Ackermann, G., Aksенov, A.A., Behsaz, B., Brennan, C., Chen, Y., et al.: American gut: an open platform for citizen science microbiome research. *mSystems* **3**(3), e00031-18 (2018)
61. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the lasso. *Ann. Stat.* **34**(3), 1436–1462 (2006)
62. Michailidis, G.: Statistical challenges in biological networks. *J. Comput. Graph. Stat.* **21**(4), 840–855 (2012)
63. Nandy, P., Hauser, A., Maathuis, M.H.: High-dimensional consistency in score-based and hybrid structure learning. *Ann. Stat.* **46**(6A), 3151–3183 (2018)
64. Paulson, J.N., Stine, O.C., Bravo, H.C., Pop, M.: Differential abundance analysis for microbial marker-gene surveys. *Nat. Methods* **10**(12), 1200–1202 (2013)
65. Pearson, K.: Mathematical contributions to the theory of evolution.—on a form of spurious correlation which may arise when indices are used in the measurement of organs. *Proc. R. Soc. Lond.* **60**(359–367), 489–498 (1897)

66. Peterson, C., Stingo, F.C., Vannucci, M.: Bayesian inference of multiple Gaussian graphical models. *J. Am. Stat. Assoc.* **110**(509), 159–174 (2015)
67. Proulx, S.R., Promislow, D.E., Phillips, P.C.: Network thinking in ecology and evolution. *Trends Ecol. Evol.* **20**(6), 345–353 (2005)
68. Ren, Z., Sun, T., Zhang, C.H., Zhou, H.H.: Asymptotic normality and optimalities in estimation of large Gaussian graphical models. *Ann. Stat.* **43**(3), 991–1026 (2015)
69. Rothman, A.J., Bickel, P.J., Levina, E., Zhu, J., et al.: Sparse permutation invariant covariance estimation. *Electron. J. Stat.* **2**, 494–515 (2008)
70. Saegusa, T., Shojaie, A.: Joint estimation of precision matrices in heterogeneous populations. *Electron. J. Stat.* **10**(1), 1341 (2016)
71. Schwager, E., Weingart, G., Bielski, C., Huttenhower, C.: CCREPE: compositionality corrected by permutation and renormalization (2014)
72. Sedaghat, N., Saegusa, T., Randolph, T., Shojaie, A.: Comparative study of computational methods for reconstructing genetic networks of cancer-related pathways. *Cancer Inf.* **13**, CIN-S13781 (2014)
73. Shojaie, A.: Differential network analysis: a statistical perspective. *Wiley Interdiscipl. Rev. Comput. Stat.* **13**, e1508 (2020)
74. Shojaie, A., Sedaghat, N.: How different are estimated genetic networks of cancer subtypes? In: Big and Complex Data Analysis, pp. 159–192. Springer, New York (2017)
75. Sondhi, A., Shojaie, A.: The reduced PC-algorithm: improved causal structure learning in large random networks. *J. Mach. Learn. Res.* **20**, 1–31 (2019)
76. Stanley, C.E., Grossmann, G., i Solvas, X.C., deMello, A.J.: Soil-on-a-chip: microfluidic platforms for environmental organismal studies. *Lab Chip* **16**(2), 228–241 (2016)
77. Tkacz, A., Hortalá, M., Poole, P.S.: Absolute quantitation of microbiota abundance in environmental samples. *Microbiome* **6**(1), 1–13 (2018)
78. Vandepitte, D., Kathagen, G., D'hoe, K., Vieira-Silva, S., Valles-Colomer, M., Sabino, J., Wang, J., Tito, R.Y., De Commer, L., Darzi, Y., et al.: Quantitative microbiome profiling links gut community variation to microbial load. *Nature* **551**(7681), 507–511 (2017)
79. Vidal, M., Cusick, M.E., Barabási, A.L.: Interactome networks and human disease. *Cell* **144**(6), 986–998 (2011)
80. Voorman, A., Shojaie, A., Witten, D.: Graph estimation with joint additive models. *Biometrika* **101**(1), 85–101 (2014)
81. Weiss, S., Van Treuren, W., Lozupone, C., Faust, K., Friedman, J., Deng, Y., Xia, L.C., Xu, Z.Z., Ursell, L., Alm, E.J., et al.: Correlation detection strategies in microbial data sets vary widely in sensitivity and precision. *ISME J.* **10**(7), 1669–1681 (2016)
82. Won, J.H., Lim, J., Kim, S.J., Rajaratnam, B.: Condition-number-regularized covariance estimation. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **75**(3), 427–450 (2013)
83. Xia, Y., Li, L.: Hypothesis testing of matrix graph model with application to brain connectivity analysis. *Biometrics* **73**(3), 780–791 (2017)
84. Xia, Y., Cai, T., Cai, T.T.: Testing differential networks with applications to detecting gene-by-gene interactions. *Biometrika* **102**(2), 247–266 (2015)
85. Xue, L., Zou, H.: Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *Ann. Stat.* **40**(5), 2541–2571 (2012)
86. Yamanishi, Y., Vert, J.P., Kanehisa, M.: Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics* **20**(suppl. 1), i363–i370 (2004)
87. Yang, E., Allen, G., Liu, Z., Ravikumar, P.K.: Graphical models via generalized linear models. In: Advances in Neural Information Processing Systems, pp. 1358–1366 (2012)
88. Yang, E., Baker, Y., Ravikumar, P., Allen, G., Liu, Z.: Mixed graphical models via exponential families. In: Artificial Intelligence and Statistics, pp. 1042–1050 (2014)
89. Yoon, G., Gaynanova, I., Müller, C.L.: Microbial networks in SPRING-Semi-parametric rank-based correlation and partial correlation estimation for quantitative microbiome data. *Front. Genet.* **10**, 516 (2019)
90. Yoon, G., Carroll, R.J., Gaynanova, I.: Sparse semiparametric canonical correlation analysis for data of mixed types. *Biometrika* **107**(3), 609–625 (2020)

91. Yu, S., Drton, M., Shojaie, A.: Generalized score matching for nonnegative data. *J. Mach. Learn. Res.* **20**(76), 1–70 (2019)
92. Yu, M., Gupta, V., Kolar, M.: Simultaneous inference for pairwise graphical models with generalized score matching. *J. Mach. Learn. Res.* **21**(91), 1–51 (2020)
93. Yuan, H., Xi, R., Chen, C., Deng, M.: Differential network analysis via lasso penalized d-trace loss. *Biometrika* **104**(4), 755–770 (2017)
94. Zhang, B., Horvath, S.: A general framework for weighted gene co-expression network analysis. *Stat. Appl. Genet. Mol. Biol.* **4**(1), 17 (2005)
95. Zhao, S.D., Cai, T.T., Li, H.: Direct estimation of differential networks. *Biometrika* **101**(2), 253–268 (2014)
96. Zhao, T., Liu, H., Roeder, K., Lafferty, J., Wasserman, L.: The huge package for high-dimensional undirected graph estimation in R. *J. Mach. Learn. Res.* **13**(1), 1059–1062 (2012)
97. Zhao, B., Wang, Y.S., Kolar, M.: Direct estimation of differential functional graphical models. In: Advances in Neural Information Processing Systems, pp. 2571–2581 (2019)
98. Zhao, S., Ottinger, S., Peck, S., Mac Donald, C., Shojaie, A.: Network differential connectivity analysis (2019). Preprint. arXiv:1909.13464
99. Zhu, Y., Shen, X., Pan, W.: Structural pursuit over multiple undirected graphs. *J. Am. Stat. Assoc.* **109**(508), 1683–1696 (2014)

Index

A

Absolute abundance, 82, 314, 319, 320, 322
Abundance, 3, 27, 46, 82, 102, 132, 175, 193,
222, 251, 272, 312
Abundance estimation, 28, 53
Actinobacteria, 134, 306–308
Adapter sequences, 47, 48, 50
Adaptive association tests, 122
Adaptive Boosting (AdaBoost), 58, 60,
65, 67
Adaptive immune responses, 134
Adaptive likelihood test, 122, 123
Adaptive MiSPU (aMiSPU), 121–124
Adaptive quadrature points, 147
AdaSyn, 63
Additive log-ratio (alr) transformation, 178,
179, 203
Additive model, 59
Ade4, 108, 109
Adiposity, 298, 304, 306, 307, 309
Adjusted Rand Index (ARI), 14–16
Aitchison distance, 103, 111, 112
Aitchison’s logistic normal distribution, 215
Albendazole, 133, 134
ALDEEx2, 179, 180
Algorithmic denoising, 5–7
Alignment-free dissimilarity measures, 91, 92
Alpha diversity, 102
Amino acid sequences, 54
AmpliCI, 7, 10, 11, 13, 14, 16–19
Amplicon clustering method, 221
AmpliconNoise, 9, 10
Amplicon Sequence Variant (ASV), 5, 175,
222, 224

Amplicon sequencing, viii, 3, 5, 12, 19, 46,
175
Amplification bias, 4, 19
Amplified PCR errors, 5, 17
ANCOM, 205
ANCOM II, 179, 180
ANOVA, 115, 179, 180
Anthelmintic treatment, 151
Antibiotic treatment, 87
Antigen, 132
Anti-resistant microbial strains, 45
APCoA, 108, 110
Arch effect, 85
Arlequin, 86
Associative analysis, 307
Asymptotic normality, 181
AUC, 39, 64, 69, 73, 296
Average-linkage clustering, 6
Average nucleotide diversity, 86
Axes of variation, 113

B

Bacterial diversity, 296
Bacterial growth dynamics, 35
Bacterial proportions, 134, 149
Bagging, 58, 61
Balanced designs, 116
Base pair similarity, 175
Bayesian algorithm, 52
Bayesian false discovery rate (BFDR), 256,
265, 267
Bayesian feature selection, 177

- Bayesian Information Criterion (BIC), 11, 186, 188, 189, 260, 320
 Bayesian methodology, 274, 285, 289
 Bayesian probabilistic clustering, 274
 Bayesian variable selection, 177, 249–269
 BBtools, 47
 BCALM2, 34
 Benchmark data set, 321
 Benjamini-Hochberg procedure, 179
 Bernoulli distribution, 252
 Beta-binomial distribution, 252, 255
 Beta distribution, 102, 252, 255
 Beta diversity, vii, 81, 82, 85, 90, 97, 101–125, 194
 Bgtools, 32
 BhGLM, 180, 181
 Bias factor, 221–227, 230–235, 237, 239–243
 Bidirectional DNA replication, 36
 Bidirectional long short-term memory, 39
 Bifidobacteria, 307, 308
 Big data, 28, 37, 296, 298, 308
 Binary cascading tree, 255
 Binary classification, 59
 Biochemical reaction order, 37
 Bioinformatics pipeline, viii, x, 46–55, 71, 73
 Bioinformatics pre-processing, 45–75
 Biomarker, 3–5, 12
 Biomarker sequencing, 3–5
 Biosynthetic gene clusters, 27, 36–40
 BiosyntheticSPAdes, 40
 Biplot, 109–113, 272, 273, 276, 277, 285, 287–289
 Bivariate split-statistics, 304
 BLAST, 90
 Body mass index (BMI), 151, 198, 204, 250, 295
 Bonferroni p-value, 237
 Boosting, 46, 58–60, 308
 Bootstrap, 58, 61, 63, 66, 72, 73, 184, 222, 302, 314–316, 329
 Bowtie2, 48, 50–52
 Branch length, 86–88, 120, 122, 123, 198
 Bray-Curtis distance, 82, 241
 Brooks Data, 223–224, 230–234, 237–240, 242, 244
 Buccal mucosa, 45, 95, 97
 Burn-in, 256, 258–260, 263, 265, 277, 281
 Burrows-Wheeler transform (BWT), 53, 54
- C**
 CAFE, 92, 95, 97
 Canberra dissimilarity, 82, 83
 Capillary sequencer, 296
- Case-control studies, 295–309
 Categorical variable, 116, 135
 Causal inference, 151
 Causation, 307
 Centered log-ratio (clr) transformation, 103, 112, 178, 179, 186, 203, 204, 210, 317, 320, 322, 325
 Chimera detection, 12
 Chi-square distribution, 118
 Chord distance, 83
 Circular consensus sequences (CCS), 11
 Clade, 28
 CLARK, 51
 Classification accuracy, 56, 58, 63
 Classification models, 56, 62, 63
 Classification tree, ix, 58, 296, 297
 Class imbalance, 56, 62–63, 65, 66, 69
 Class membership, 57, 58
 Cluster centroids, 6
ClusterFinder, 38–40
 Clustering, 5, 6, 8–10, 14–19, 32, 84, 95, 97, 102, 111, 114, 115, 221, 274, 287
 Clusters, 5–10, 13, 15–18, 27, 33, 36–38, 55, 73, 84, 85, 97, 109, 112, 114, 115, 121, 124, 132, 241, 272
 COCACOLA, 93
 Coincidence, 103, 275
 COMBO dataset, 198, 204
 Community diversity, 3, 108
 Compacted de Bruijn graph, 33–34, 41
 Complete-linkage clustering, 6, 10
 Completeness, 6, 15, 16
 Complex missingness, 229
 Compositional data, 29, 57, 131–171, 178–181, 183, 189, 203–204, 206, 216, 250–251, 257, 311–332
 Compositional mediation model (CMM), 184
 Compositional mediators, 184
 Compositional methods, 178–180
 Compositional model, 222
 Computational time, 151, 329
 CONCOCT, 32, 33
 Conditional entropy, 15
 Conditional probability, 300–302
 Conditional unweighted split-statistic, 301
 Conditional weighted split-statistic, 299, 301
 Condition Gini index, 300–302
 Confounding, 82, 103, 108, 151, 177, 185
 Conjugate, viii, 133, 135, 138, 139, 149, 202, 251, 254
 Consensus sequences, 5, 11
 Constraining variable, 106
 Contaminants, 11, 15, 19, 47
 Contamination, 9, 12, 14, 19, 47, 216

- Contigs, 30–37, 40
Coordinate descent algorithm, 207
Copula models, 216, 320, 322, 323, 325, 326
Correlation, ix, viii, 57–59, 83, 102, 111, 124, 131–133, 135, 139, 141–144, 148–150, 168–171, 186–188, 196, 228, 229, 254, 257–259, 261, 274, 282, 300, 307, 312–331
Count-based methods, 180
Covariance matrix, 136, 142, 156, 158, 161, 178, 228, 229, 235, 241, 244, 245, 273, 289, 318, 322, 323
Covariate adjustment, 185–188
Covariates, viii, 104, 108, 113, 114, 125, 131–136, 139, 141–144, 146, 148–152, 155, 165, 179, 185, 186, 189, 195, 196, 198, 222, 225, 227, 230–232, 241, 243, 249–253, 255–261, 263–267, 269, 289
Coverage number, 33
Covering Point Sets (CPS), 17, 18
CovTools, 245
Credible intervals, 273, 276, 282–286, 289
Critical Assessment of Massive Data Analysis (CAMDA), 46, 60, 65, 73, 272, 288, 289
Crohn’s disease, 36, 37
Cross-sample consistency, 5
Cross-sectional study, 198
Cross-validation, 56, 61–63, 286, 317, 318
Cutadapt, 47
Cycle nucleotide, 9
Cytokine response, 132–134, 144, 147, 148, 150, 151
- D**
DADA2, 7–11, 13, 14, 16–19
Data augmentation techniques, 63
Data visualization, viii
DAtest, 181
De-biased LASSO, 183
Deblur, 6, 7, 9, 11, 19
De Bruijn graphs, 30–34, 40, 41
Decision boundaries, 59
Decision tree, 58
Decomposition, ix, 229, 273–275, 289
Deduplication, 221
DeepBGC, 38–40
Deep learning, 39, 59
DEMIC, 39, 40
Denoising, 3–19
DESeq, 180, 205
DESeq2, 205
DIAMOND, 51
- Dice coefficient, 82, 84, 103
Dichotomous features, 58, 295
Diet, 151, 198, 205, 262, 307
Differential abundance, 176–182, 189, 194, 205, 216, 241
Differential bias, 223
Differential expression analysis, 176
Diffuse prior, 252, 254
Dimension reduction, 33, 61, 66, 106, 194, 213, 278, 286
Dirac-delta, 252
Directed graph, 30
Directional, 121, 277
Dirichlet distribution, 195, 196, 202, 251, 321
Dirichlet mixture prior, 215
Dirichlet-multinomial mixed model, 138, 139, 143, 148, 155, 168, 169
Dirichlet-multinomial model, 195, 216, 254
Dirichlet prior, 202, 215
Dirichlet-tree multinomial distribution, 196, 254
Dirichlet-tree multinomial model, 196, 254, 267
Disease association studies, 27
Dissimilarity matrix, 82, 95, 106, 115, 194, 278
Distance-based analysis, 101–125, 183, 241
Distance-based longitudinal microbiome association studies, 125
Distances, ix, viii, 5–7, 10, 16, 32, 59, 82–85, 87, 94, 98, 101–125, 183, 194, 198, 241, 271–290
DNA extraction, 3, 221, 222, 225, 237, 241
DNA sequencing platform, 3
DNA synthesis reaction, 9
Double principal coordinate analysis (DPCoA), 108–110
Down-sampling techniques, 62
DSRBF, 63
Duality, ix, 271–290
Dummy outcome variable, 161
- E**
Earth Microbiome Project (EMP), 46
Ecological research, 81
Ecological studies, 272
Ecoreg, 138, 159
ECP property, 304
Edge, 30, 31, 33, 40, 59, 253, 254, 259, 265, 298, 299, 304, 306, 313, 318, 322, 329, 331
EdgeR, 180, 181
Eigendecomposition, 106

Eigenmatrix, 274, 275
Eigenvalues, 107, 118, 281, 322
ElasticNet, 57
Empirical Bayes, ix, 201–202, 204–206, 245
Entropy, 15, 184, 296
Environmental conditions, 176, 189
Environmental exposures, ix, 298, 306, 307, 309
Environmental factors, 60, 194, 304, 309
Environmental gradient, 82, 84, 85, 92, 93
Environmental microbiome analysis, 45
Environmental perturbations, 5
Error model misspecification, 14
Euclidean distance, 82, 83, 98, 103, 108, 111, 112, 115, 118
Eulerian walks, 30–32
Evolutionary relationships, 85, 86, 90, 193, 196, 254
Exchangeability, 116
Expectation-maximization (EM) algorithm, 10, 181
Experimental perturbations, 5
Exponentially decay, 10
Extreme Gradient Boosting (XGBoost), 46, 59, 60, 65–68, 72

F

False negative rate, 257
False positive rate (FPR), 5, 257, 323–325
Fasta files, 97
FastQC, 47–50
Fastq files, 46, 54, 55
Feature identification, viii, 175–189
Feature reduction, 56, 286
Feature selection, 56–58, 60, 65, 183, 190, 296
Fecal metagenomic samples, 93
Finite mixture model, 9–11
Fixed effects, 142, 143, 147, 155–157, 161, 163, 167
Fixed order Markov chain (FOMC), 92
Fizzy, 57
Flowgram, 9, 10
Fluorescent intensity, 9
F-measure, 64
FM-index, 54
Forensic studies, 45
Frobenius norm, 230, 274–276, 278, 281
F-statistic, 228, 229
F-test, ix, 179
Full-length gene, 11
Full-rank model, 226, 229
Functional microbiome, 36

G

Gauss.hermite, 138, 158
Gauss–Hermite quadrature, 133, 137, 138, 142, 147, 157
Gaussian graphical model, 253, 258, 269
Gaussian mixture model, 33
Gauss Noise, 66
Gene content profiling, 36
Gene expression data, 180
Gene expression studies, 27, 176, 181
Gene profiling data, 81, 82, 90, 97
Genera, 3, 262, 267
Generalized linear models, 102, 113, 125, 136, 186, 216
Generalized UniFrac, 87, 88, 94, 103, 104
Genetic diversity differences, 86
Genome copy number, 35
Geographical location, 46, 82
Geographical origin, 287, 289
GGally, 186, 188, 328
Gibbs sampler, 255, 277–278, 280
Gini index, 296–302, 308
Global dissimilarity, 101, 109
Global minimum, 275
G-mean, 64, 66–68
Goodnes-of-fit, 135, 141, 150, 210
GOTTCCHA, 51
Gower distance, 83, 84, 272, 274, 286, 287, 289
Gradient boosting machines (GBM), 59, 111
Gradient descent optimization, 59
Gram–Schmidt orthogonalization/process, 116, 281
Graphical structure, 252, 256–259, 262–265, 269
Graph neighborhoods, 34, 253
Greengenes, 285, 287
Growth dynamics, 35
GUUniFrac, 94, 95, 104, 105
Gut microbial community, 132, 133

H

Haldane-like correction, 225, 236
Hamming dissimilarity, 84
Hat matrix, 115, 118, 279
HDeconometrics, 186, 188
Hellinger distance, 83
Helminth-endemic area, 132, 133
Helminth infection, 132, 134, 148, 150
Heterogeneity, 132, 148, 249
Heterogeneous dispersion, 116
Heteroscedasticity, 242, 319
Hierarchical Bayesian models, 250

- Hierarchical divisive clustering, 8–9
Hierarchical feature engineering (HFE), 57, 66, 68, 69
Hierarchical prior, 253
High-dimensional, 36, 56, 58, 102, 113, 117, 150, 175, 182, 184, 189, 194, 206, 250, 273, 282, 295, 312, 319
High-dimensional objective function, 273
High throughput sequencing, 3, 90, 95, 249
HiSeq, 46, 221
HMMER, 40
Hmmscan, 38
HMP2Data, 262
Hoff algorithm, 277
Homogeneity, 15, 16, 211
Homopolymer, 9, 10
Host reads, 47
Human intestinal microbial communities, 86–88
Human microbiome, 28, 36, 45, 46, 249, 261, 295
Human Microbiome Project (HMP), 28, 46, 152, 261
Hybrid techniques, 63
Hypergeometric distribution, 89
Hypervariable regions, 3, 4
Hypothesis testing, viii, 102, 104, 105, 109, 113–124, 179, 181, 196, 239
- I**
Identifiability, 135, 136, 139, 209, 210, 227
Identity of indiscernibles, 103
Illumina, 5, 7, 10, 11, 16, 46
Illumina HiSeq next-generation shotgun sequencing, 46
Immune response, 132, 134
Immunological examination, 133
Importance scores, 56, 121
Impurity, 296, 297
Imputation, 243
Indel error, 11
Indonesia, 132, 133, 148
Infection status, 132, 144, 148, 151
Inflammatory bowel diseases (IBD), 88
Informative variables, 300, 301
Innate immune responses, 134
Integrative Human Microbiome Project, 28
Interaction effect, 186, 224, 230–232, 234–239, 242
Inter-cluster distance, 6
Interleukin-10 (IL-10) response, 132, 134, 144
IRep, 40
- Isometric log-ratio (ilr) transformation, 111, 112, 178
Isometry, 104, 178, 183
- J**
Jaccard distance, 82, 103
JELLYFISH, 41
Joint marginal distribution, 137
- K**
Kaiju (KJ), viii, 51, 53–56, 65–67, 71
KEGG orthology pathways, 250
Kernel density, 184
Kernel machine regression, 117–119, 122, 123
Kernel trick, 57, 59
k-medoid clustering, 32
k-mer counts, 93
k-mer matching, 29
KneadData, 48, 50
K-nearest neighbor, 286
Kraken, viii, 29, 41, 51–53, 55, 93
Kraken2, viii, 51–53, 56, 65, 67–71
Kraken2-bracken system, 53, 56, 65, 69
Kruskal-Wallis, 179
Kullback-Leibler divergence, 92
- L**
LASSO, 183, 186, 188, 189
Lasso, 57, 206–209, 212–216, 260, 318, 321, 323, 326
Latent inclusion, 250, 253
Latent variable model, 216
Latent variables, 102, 113, 216
Least common ancestor (LCA), 29
Least squares, 65, 183, 206, 210, 226, 228, 242, 243
Leaves, 14, 89, 196, 197, 211, 296
Lennon dissimilarity, 84
Levenshtein distance, 7
Library preparation, 3, 5
Linear classifiers, 57
Linear discriminant analysis, 57
Linear log-contrast model, 206–209, 213
Linear mixed model, 118, 125, 135, 149
Linear regression, 108, 186, 212–215
Linkage hierarchical clustering, 10
Link function, 186, 198
Lipopolysaccharide (LPS), 132, 134, 144
Local minimum, 289
Local optimum, 10
LOESS regression, 11

- Logistic, 57, 135–137, 143, 144, 148, 155, 161, 170, 171, 215, 298, 323, 326
 Logistic regression, 57
 Logit-dependent random effect, 141
 Log-linear model, ix, 135, 221–245
 Log-ratio LASSO, 206–209
 Log-ratio transformation, 103, 178, 179, 204, 210, 216
 Log transformation, 186
 Longitudinal design, 132
 Long Read Technology, 11–12
 Loss function, 273, 274
 Lower-dimensional space, 106
 Lowest common ancestor, 52
 Low-prevalence samples, 271
 Low-quality reads, 47
 Lsei, 228
- M**
- Machine learning, 28, 41, 56, 57, 60–62, 64, 65, 71, 190, 206, 308
 MAFFT, 7
 MAGHIT, 30
 Main effect, 224, 230–237, 239–241
 Majority voting, 58, 59, 62
 Mammalian gut microbial communities, 88
 Marginal correlations, 135, 141, 143, 144, 148–150, 168–171, 186
 Marginal Pearson correlations, 150
 Marginal posterior probability of inclusion (MPPI), 256, 257, 263, 265–267
 Markov chain Monte Carlo (MCMC), 250, 255–258, 260, 261, 265, 273, 275, 277, 280, 281, 287
 Markov model, 38–39, 91–93
 Markov random field (MRF), 253
 MASS, 95, 181, 194, 198, 250, 295
 Matthew’s correlation coefficient (MCC), 257–261
 MAUC, 64, 66–68
 Maximum likelihood, 133, 181, 196, 199, 201, 203, 206
 MBMC, 93
 ‘McLaren, Willis and Callahan (MWC) model,’ 40, 221–225, 241, 243
 McNemar’s test, 295
 Mean abundance level, 178
 Mean shifts, 108, 113
 Mean-variance relationship, 102
 Measure of similarity, 273
 Mediation analysis, viii, 177, 182–184, 241
 Mediation effects, 177, 182–184
 Mediator, 182–184
 MetaBAT, 93
 MetaBAT2, 30, 32, 33
 Metabolomic potentials, 28
 Metadata, 105, 193, 199, 200, 208
 Metagenome-assembled genomes (MAGs), 30, 33, 36
 Metagenome assemblies, 30–36, 40
 MetagenomeSeq, 53, 180, 181, 205
 Metagenome structural variants, 27
 Metagenomic forensics, 45–75
 Metagenomic sequencing assemblies, 28
 MetaPhlAn2 (MP), viii, 28, 51, 52, 56, 65–67, 71
 MetaSUB, 46, 48, 272, 288
 Metataxonomics, 46
 Metatranscriptomic datasets, 92
 Metatranscriptomics, 92, 175
 Metric, 14, 16, 51, 61, 64, 71, 72, 102, 103, 106, 107, 124, 183, 194, 202, 296
 Metropolis-Hastings, 255
 Microbial cells, 81
 Microbial communities, viii, 3, 12, 27, 28, 32, 45, 46, 81–90, 97, 98, 103, 108, 132, 133, 193, 201, 210, 216, 272, 295, 311
 Microbial composition, vii, 92, 101, 201–206, 249
 Microbial features, 27, 176
 Microbial genes, 295
 Microbiome analysis, 45, 102, 271–290
 Microbiome based sum of powered score (MiSPU), 120–124
 Microbiome comprehensive association mapping (MiCAM) tests, 124
 Microbiome features, viii, 106, 107, 175–182, 184–187, 189, 295
 Microbiome fingerprint, 60, 287, 289
 Microbiome regression-based kernel association tests (MiRKAT), 114, 117–120, 122–124
 Microbiome Study-Pregnancy Initiative, 250, 261–264
 Microbiota, 36, 60, 92, 296, 307
 MicroBVS, 250, 261, 262, 264, 268
 Midpoint rooting, 217
 MiLineage, 180
 Minimum Information about a Biosynthetic Gene Cluster (MIBiG) database, 37, 38, 41
 Misclassification, 62, 302
 MiSeq, 221
 Missing data, 58, 222, 241
 Mixed effect Dirichlet-multinomial model, 135
 Mixed effect multinomial logistic model, 135
 Mixed effects model, 139, 150, 179

- Mock community data, 223, 224, 241, 243
Model-Based Ordination, 113
Model-based standardization, 186, 188–189
Monte Carlo hypothesis, 228
Monte Carlo sampler/sampling, 141, 179, 277
Mothur, 6, 271
mOTU, 28, 51
Multiclass classification problems, 57, 59, 64
Multidimensional scaling (MDS), 95, 106
Multilayer perceptron (MLP), 58, 59, 66–68
Multilevel models, 295
Multinomial count outcomes, 142, 143
Multinomial distribution, 196, 233, 236, 251, 254
Multinomial logistic mixed model (MLMM), 136–143, 146–150, 165, 166
Multinomial model, 57, 195–198, 216, 254–255, 267
Multi-omics, 125, 250, 261–264
Multiple comparisons, 101, 124, 179, 237
Multiple-group comparisons, 181
Multiple univariate regression model, 131
Multiplex real-time PCR, 133
Multiplicity, 256
Multiscale dimension reduction, 213
Multiscale subcomposition method, 210
Multivariate count data, 194–200, 216, 260
Multivariate count outcome, 135, 137, 148, 151, 155, 161
Multivariate exposures variable, 183, 184
Multivariate Gauss-Hermite quadrature, 137, 138, 158
Multivariate Gini index, 299–301
Multivariate modeling, 102
Multivariate Poisson-lognormal distribution, 216
Multi-way relationships, 190
Mutation rate, 54
- N**
Nature language processing (NLP), 39
Negative binomial model, 180, 181
Network prior, 252–254
Neural network, 46, 58, 60, 216
Newton boosting, 59
Newton–Raphson, 59
Next generation sequencing (NGS), 45–47, 92, 93, 97, 175
Neyman–Pearson feature selection, 57
Niche, 102
Nitrogen fertilizer, 185
Nitrogen fixing bacteria, 176
Noise variables, 298–301, 303, 308
Non-additive relationships, 184
Non-identifiability, 227
Non-informative prior, 252
Non-linear classifiers, 57
Nonlinear relationships, 184
Nonparametric Entropy Mediation (NPEM), 184
Nonparametric omnibus test, 183
Non-phylogenetic measures, 103
Non-ribosomal peptide synthetase (NRPS), 37
Normal distribution, 136, 138, 146, 149, 179, 181, 252, 258, 277, 318–320, 322, 323
Normalization, 16, 33, 35, 53, 181, 201–205, 216, 222, 225, 312, 314, 315, 317, 325, 327
Normalization approaches, 201
Nucleic acid databases, 90
Nucleotide differences, 3, 13, 86
Nucleotide sequences, 5, 46, 54
Null hypothesis, 8, 86, 89, 113, 115, 116, 118, 120, 150, 177, 227, 228, 235
- O**
Obesity, 295, 296, 298, 304–309
Ochiai coefficient, 84
Operational Taxonomic Units (OTUs), ix, 5, 6, 16, 57, 63, 64, 82–85, 87, 90, 94, 104, 109, 111, 114, 115, 175, 186–189, 193, 194, 196, 198, 201, 202, 205, 206, 214, 222, 224, 231, 232, 242, 262, 264, 267, 271–290, 296, 307
Optimal ensemble classification algorithm, 46
Optimal microbiome-based association test (OMiAT), 122–124
Optimal microbiome-based survival analysis (OMiSA), 122
Optimal MiRKAT (OMiRKAT), 118, 119, 122–124
Optimal split, 298, 299, 304, 306, 307
Ordinary least squares, 183
Ordination analysis, 102, 106, 216
Orthonormal, ix, 116, 274–277, 279, 281, 289, 290
OTclust, 17
OTU identifiers, 109
Out-of-bag, 58, 302
Overdispersion, viii, 9, 11, 102, 133, 135, 138, 139, 142, 148, 156, 157, 163, 180, 251, 257
Overparameterized model, 208, 228, 229
Over-sampling techniques, 62, 63, 69
Oxford Nanopore Technology, 11

P

Pacific Bioscience, 11
 Paired data analysis, 307, 308
 Paired-end WGS reads, 47
 Pairwise distances, ix, 5, 6, 16, 183
 Pairwise fused lasso, 214
 Pairwise similarity, 117
 Parallel computing, 59, 73
 Parameter tuning, 58, 207, 260, 317–321, 328, 329
 Parsimony score, 86
 Patristic distance, 108
 PCR bias, 14, 221
 Peak-to-trough ratio, 35
 Pearson correlation coefficient, 312
 Penalized approach, 250, 259–261
 Penalized likelihood, 196, 249, 250, 319, 320
 Penalized regression, 57, 260, 261, 264
 Penalty parameter, 186
 PERMANOVA-S ensemble test, 116
 Permutation, ix, 36, 109, 110, 112, 115–123, 184, 222, 228, 229, 235, 241, 242, 301, 314
 Permutational multivariate analysis of variance (PERMANOVA), 115–117, 119, 123, 124
 Permutation-based, ix, 123
 Permutation *p*-values, 110, 112, 115
 Permutation tests, 115–117
 Permutated monotone matrix model, 35, 40
 Pfam2vec, 39
 Pfam domain sequential order, 38, 39
 Phenotypes, viii, 33, 36, 60, 101, 113, 114, 125, 194, 196, 209, 213, 311, 312
 PhILR transformation, 111, 112
 Phred scores, 47–49
 pH values, 84
 Phylogenetic convolutional neural networks, 216
 Phylogenetic structure, 108, 250, 268
 Phylogenetic test statistic, 85–86
 Phylogenetic tree, viii, 85–90, 94, 104, 114, 121, 124, 193, 196, 198, 199, 214, 217, 244, 254, 265, 267, 269, 272
 PhyloMDA, 198–200, 204, 212, 215
 Phyloseq, 108, 181, 198, 199, 264, 321
 Physiologic states, 295
 Placebo, 133, 134, 151
 Poisson, 8, 9, 11, 181, 199, 215, 216
 Poisson distribution, 8, 181
 Poisson graphical model, 215
 Poisson-multinomial distribution, 216
 Polyketide synthase (PKS), 37

Polymerase chain reaction (PCR), 4, 5, 9, 12, 14, 16, 17, 19, 133, 221–224, 237, 238, 240
 Population differentiation statistic, 86
 Positive predictive value (PPV), 64, 69, 70
 Post-burn-in, 277, 281
 Posterior, 39, 201–203, 250, 255–256, 263, 266, 273–283, 285, 286, 289, 316, 325
Post hoc analyses, 106
 Precision, 9, 12–15, 39, 64, 190, 242, 244, 245, 253, 275, 278, 281, 286, 300
 Precision matrix, 253, 322
 Prediction error, 297, 301, 302
 Predictive power, 297
 Primer IDs, 16
 Principal component analysis (PCA), 33, 35–37, 65–68, 111–113, 296
 Principal components (PCs), 65, 95, 107, 111, 113, 271–290
 Principal coordinates analysis (PCoA), 95, 96, 106–110, 112, 113, 194
 Probabilistic denoising, 5
 Proc NLMIXED, 146, 147, 150, 151, 161, 163
Prodigal, 38
 Profile hidden Markov models, 38–39
 Profile taxonomic composition, 27
 Projection matrix, 108, 115, 118, 245
 Prokaryotes, 28
 Proportion of mapped reads, 51, 55
 ProWsyn, 63
 Pscl, 181
 PyroNoise, 9
 454 pyrosequencing, 9, 10
 Pyrosequencing, 9, 10, 134, 198
 Python, 63, 95

Q

QIIME, 6, 198, 271
 Quadratic discriminant analysis, 57
 Quality control, 46–51, 54, 71, 193, 272
 Quality scores, 7, 8, 10, 11, 14, 48, 49

R

Rand index (RI), 14
 Random effects, viii, 102, 118, 131–133, 135–137, 139–143, 146–148, 150, 151, 155, 161, 167
 Random feature selection, 296
 Random forest (RF), 46, 58, 60, 63, 66–68, 70–72, 296, 308
 Randomized controlled trial, 133

- Rarefaction, 103, 111, 181, 321
Rarefying, 201, 205, 216
Ratio transformations, 203
Raw sequencing data, 40
Rcpp, 250
RcppArmadillo, 250
Read filtering, 12, 221
Read mapping, 47, 51, 221
Recall, 12, 14, 15, 39, 64, 231, 239, 304
Recursive partitioning (RPart), 58, 65–68
Recursive splitting, 58
Reduced feature space, 56, 68, 69
Reference-based methods, 6
Reference database, 6, 16, 19, 53, 54
Reference-free (*de novo*) methods, 5, 91
Reference protein database, 53
Regression, ix, 11, 57, 108, 117–119, 123, 131, 136, 139, 163, 177, 179, 184, 186, 189, 194, 196, 198–200, 204, 206–215, 226, 241, 249–269
Regression-based kernel association tests, 117
Regularized discriminant analysis, 57
Relative abundance, viii, ix, 5, 6, 9, 11, 28–30, 35, 40, 51, 52, 82, 84, 97, 98, 103, 104, 107, 111, 132, 194, 201–203, 205, 206, 216, 222–224, 232–234, 240–243, 272, 312, 314, 315, 319, 320, 323, 330
Relative bias, 231, 242
Replication rates, 35, 36, 40
Reproducing kernel Hilbert space, 117
Rhizosphere microbiome, 176, 185
RNA-sequencing (RNA-seq), 180, 181, 216
ROC, 296, 323, 325, 326
Root node, 58, 211, 255, 267, 296, 297
Root-to-leaf (RTL) path, 29
16S rRNA, viii, 3, 4, 9, 16, 27, 29, 46, 81, 82, 84, 90, 92, 94, 97, 132, 134, 193, 262, 271, 296, 311
Run time, 51, 54
- S**
Sample clustering, 84, 287
Sample volume, 103
SAS, 146, 147, 150, 151, 161, 163, 167
Saturated log-linear model, 135
Scatterplot, 95, 109
Scree plot, 273, 276, 285, 286, 289
Sensitivity, 3, 9, 14, 51, 54, 64, 250, 257–261, 322, 323
Sensitivity analysis, 250, 252, 258–259, 261
Sequence similarity, 5, 6
Sequence similarity threshold, 5
Sequencing depth, 12, 84, 85, 90, 93, 201, 224, 225, 321, 322
Sequencing platforms, 3, 194, 201, 221
Shared effects, 135, 136, 139, 141, 143, 145, 148, 150, 165, 166, 168–171
Shotgun metagenomics, viii, 3, 27–41, 97
Signal variables, 301, 303
Simple linear growth model, 35
Simplex, 195, 202, 203, 212, 251
Single-linkage clustering, 6
Single nucleotide polymorphisms (SNPs), 27
Single-nucleotide sequence resolution, 5
Singular value decomposition, ix, 229, 273
Singular vectors, 36, 40, 273
Skinny bayesian technique, 273, 274, 278–280, 285
Smith–Waterman algorithm, 90
SMOTE, 63
Smote-variants, 63
Sorensen–Dice coefficient, 105
Sparse Microbial Causal Mediation Model (SparseMCMM), 183, 184
Spearman correlation coefficient, 83
Spearman’s rank correlation, 186, 187
Species abundance matrix, 56
Species composition, 102
Species-level genome bins (SGBs), 33
Specificity, 257–261, 322, 323
Spike-and-slab priors, ix, 250–252
Spike-in controls, 19
Split-statistic, ix, 297–308
Splitting point, 301
Squared error loss function, 273
Standardization, 185, 186, 188–189
Statistical power, 102, 123, 131, 233
Stiefel manifold, 275, 279, 281
Stochastic distances, 272
Stochastic functions, 272
Stool, 45, 95, 97, 133, 198, 296
Strain identification, 34
Strain-level analysis, 28, 33
Structural information, 254
Structural missingness, 222–224, 229
Subcomposition analysis, 209
Substitution probability, 11
Sum of powered score tests, 120, 122, 123
Sum-to-zero constraints, 57
Supervised classification, 57, 71
Supervised classifiers, 46, 71
Support Vector Machines (SVMs), 46, 57–60, 66–68
Supragingival plaque, 95, 97
Symmetry, 103

T

- Taxa, 41, 51, 101, 175, 193, 221, 250, 312
- Taxonomic assignment, 46, 51, 55
- Taxonomic identifiers, 109
- Taxonomic profiling, 12, 29, 51–56, 61, 65, 67–71
- Taxonomic tree, 29, 41
- Taxonomy table, 57
- Taxon-taxon interaction, 222, 225, 237, 238, 241
- Template nucleotide, 9
- Terminal node, 58, 59, 196, 296
- Tetranucleotide frequency, 32
- Three-way relationships, 151, 190
- Tidyverse, 54
- Tongue dorsum, 95, 97
- Total bacterial concentration, 103
- Total sum scaling (TSS), 201, 205
- Training data, 39, 57, 58, 61, 69, 70, 72, 73
- Transcriptomic datasets, 92
- Transition probability matrix, 91
- Treatment conditions, 179
- “Tree-guided Automatic Subcomposition Selection Operator (TASSO)”, 212
- Tree impurity, 297
- Tree splitting, ix, 296, 297, 308
- Triangle inequality, 103, 272
- Trimming, 12, 48, 221
- Trimmomatic, 47–50
- T-SNE, 17, 18
- T*-test, 205, 206
- Tuning parameter, 207, 260, 317–321, 329
- Two-group comparisons, 181
- Type I error, 114, 118, 235, 242

U

- UBL, 63
- Unconstrained ordination analysis, 102
- Undirected graphical models, 253
- Unequal variances *t*-test, 179
- Uniform distribution, 298
- Uniform prior, 202, 316
- UniFrac, viii, ix, 82, 87–88, 90, 94–97, 103–110, 112, 114, 116, 117, 119, 120, 123, 124, 241, 272, 274, 286, 287, 289
- Unique molecular identifiers, 12, 16, 19
- Unitigs, 30, 33, 34
- Universal marker genes, 28
- Universal primer, 4
- UNOISE2, 6, 7, 9
- UNOISE3, 13, 14, 16, 19

UPARSE, 6

- Urban environments, 45

V

- Variable fusion, ix, 212–215
- Variable importance, 58, 70, 301–304
- Variable length Markov chain (VLMC), 92
- Variable ranking method, 297, 299, 303, 307, 309
- Variable regions, 3, 4, 193
- Variable selection, 54, 57, 177, 185, 186, 189, 249–269, 295–309
- Variance adjusted weighted UniFrac (VAW-UniFrac), 82, 87–90, 94–96, 103
- Variance component score test, 117, 118
- Variational Bayesian approximation, 33
- Variscan, 86
- Visual separation, 107
- V-measure, 15, 16
- Von Mises–Fisher matrix distribution, 277, 278, 280

W

- Wald test, 150
- Weak classifiers, 58, 59
- Weighted rank aggregation, 61, 66, 73
- Weighted UniFrac (W-UniFrac), 82, 87–90, 94–96, 103–110, 112, 117, 119, 123, 241
- Weighting, 62, 63, 69, 72, 120, 122, 242
- Whole blood cytokine responses, 132, 133
- Whole genome sequencing (WGS), 47, 49, 51, 65, 71, 73
- Whole-metagenome shotgun (WMS), 175
- Wilcoxon rank test, 179
- Within-sample diversity, 102
- Word pattern occurrences, 82

Z

- Zero-hurdle model, 180, 181
- Zero-inflated beta regression (ZIBR) model, 179, 180
- Zero-inflated generalized Dirichlet-multinomial (ZIGDM) model, 180, 216
- Zero-inflated normal model, 181
- Zero-inflated overdispersed Poisson distribution, 181
- Zero-inflation, 180, 181, 216
- ZIBseq, 179, 180