UNIVERSITY OF PADOVA
DEPARTMENT OF INFORMATION ENGINEERING

Ph.D. Course in Information Engineering
Curriculum: Bioengineering
Series: XXXI

16S rRNA gene sequencing sparse count matrices: a count data simulator and optimal pre-processing pipelines

Ph.D. Candidate: Ilaria PATUZZI

Advisor: Prof. Barbara DI CAMILLO

Co-Advisor: Giacomo BARUZZO

Coordinator: Prof. Andrea NEVIANI

November 2018

To my loving husband, without the love, help and support of whom this work would never have been realized.

Ringraziamenti

Vorrei dedicare qualche riga al ringraziamento di tutte quelle persone che, con la loro presenza, la loro vicinanza ed il loro sostegno, hanno contribuito a rendere possibile questo nuovo traguardo.

Il mio primo pensiero va a mio marito, Alessandro, che con amore e pazienza ha saputo starmi accanto anche nei momenti più difficili. Il suo inesauribile ottimismo e la sua profonda fiducia sono stati linfa vitale per affrontare l'impegno e gli ostacoli di questo percorso.

Il secondo, ma non per importanza, va alla mia famiglia. A mia madre Isabella e mio padre Umberto, che hanno creduto in me anche nei momenti in cui io stessa dubitavo e mi hanno insegnato a puntare sempre in alto. A mia sorella, Federica, per essere stata sempre presente con il suo amore ed il suo sostegno. A mia nonna Fanny, che mi ha insegnato con la sua vita che non esistono imprese impossibili. A mio nonno Gerardo, che porto sempre nel cuore, e mia nonna Grazietta, per il loro incondizionato affetto.

Un sentito ringraziamento va al mio supervisore, Barbara, per aver reso questi tre anni di ricerca insieme una fonte continua di crescita culturale, professionale e personale e per aver messo a disposizione di questo progetto la sua profonda competenza e la sua umanità.

Un grandissimo grazie ai membri del mio gruppo di ricerca Alberto, Alessandra, Alessandro, Erica e Giacomo, per avermi accolta a braccia aperte fin dal primo istante e per aver reso così piacevole e divertente questo periodo così impegnativo. In loro ho trovato non solo grande competenza, ma anche comprensione, sostegno, affetto ed amicizia, senza i quali questo mio percorso non avrebbe certamente avuto lo stesso valore. Un ringraziamento speciale a Giacomo per i consigli preziosi ed il tempo dedicatomi, grazie ai quali ho potuto imparare e migliorare moltissimo.

Un grazie, inoltre, agli amici e colleghi Agnese, Alberto, Alessandra, Alessandro P., Alessandro Z., Alessia, Enrico, Erica, Giada, Ilaria, Marco, Maria, Martina B., Martina V., Michele, Roberto e Simone per le pause pranzo spensierate ed i bei momenti trascorsi insieme. Un sincero ringraziamento a Sergio, per avermi incessantemente spronata e motivata in questi ultimi mesi.

Un grande ringraziamento a tutti i colleghi dell'IZSVe, che hanno condiviso con me i progetti, i successi e le incertezze di questi tre anni di studio. Grazie alle ragazze della Stanza

1, Alessandra, Eleonora, Marzia e Sara, per avermi aiutata sostenendomi, giorno dopo giorno, con la loro vicinanza e la loro allegria. Un ringraziamento particolare ai miei responsabili, Antonia, Carmen e Lisa, che hanno reso possibile questo Dottorato.

Grazie, infine, a tutti gli amici che mi sono stati vicini in questi anni e che, con il loro affetto, la loro spensieratezza e la loro gioia mi hanno aiutata a raggiungere questo obiettivo.

Abstract

The study of microbial communities has deeply changed since it was first introduced in the 17th century. When the pivotal role of microbes in regulating and causing human diseases became evident, researchers began to develop a variety of techniques for isolating and growing microbes in the laboratory, with the aim of characterizing and classifying them. In the late 1970s, a breakthrough in the way bacterial communities were studied was brought by the discovery that ribosomal RNA (rRNA) genes could be used as molecular markers to perform organisms classification. Some decades later, the advent of DNA sequencing technology revolutionized the study of microbial communities, permitting a culture-independent view on the overall community contained within a sample.

Today, one of the most widely used approaches for microbial communities profiling is based on the sequencing of the gene that codes for the 16S subunit of prokaryotic ribosome (16S rRNA gene). As ribosome plays an essential role in prokaryotic life, it is ubiquitous to all bacteria, but its exact DNA sequence is unique to each species. For this reason, it is used as a sort of molecular fingerprint for assigning to each community member a taxonomic characterization.

The advent of Next-Generation Sequencing (NGS) platforms, able to produce a huge amount of data reducing the related time and costs, ensured 16S rRNA gene sequencing (16S rDNA-Seq) an increasing growth in election rate as preferred methodology to perform microbiome studies. Despite this, the continuous development of both experimental and computational procedures for 16S rDNA-Seq caused an unavoidable lack in standardization concerning sequencing output data treatment and analysis. This is further complicated by the very peculiar characteristics that distinguish the matrix in which samples information is summarized after sequencing. In fact, the instrumental limit on the maximum number of obtainable sequences makes 16S rDNA-Seq data compositional, i.e. they are data in which the detected abundance of each bacterial species is dependent from the level of presence of other populations in the sample. Additionally, 16S rDNA-Seq-derived matrices are typically highly sparse (70-95% of null values). This is due both to biological diversity between samples and to the loss of information about rare species during sequencing, an effect that is heavily dependent on the usually skewed distribution of species abundances internal

to microbiomes and on the number of samples sequenced in the same sequence run (the so-called multiplexing level).

The above peculiarities make the commonly adopted loan of bulk RNA sequencing tools and approaches inappropriate for 16S rDNA-Seq count matrices analyses. In particular, unspecific pre-processing steps, such as normalization, risk to introduce biases in case of highly sparse matrices.

The main objective of this thesis was to identify optimal pipelines that filled the above gaps in order to assure solid and reliable conclusions from 16S rDNA-Seq data analyses. Among all the analysis steps included in a typical pipeline, this project was focused on the pre-processing of count data matrices obtained from 16S rDNA-Seq experiments. This task was carried out through several steps. First, state of the art methods for 16S rDNA-Seq count data pre-processing were identified performing a thorough literature search, which revealed a minimal availability of specific tools and the complete lack in the usual 16S rDNA-Seq analysis pipeline of a pre-processing step in which the information loss due to sequencing is recovered (zero-imputation). At the same time, the literature search highlighted that no specific simulators were available to directly obtain synthetic 16S rDNA-Seq count data on which to perform the analysis to identify optimal pre-processing pipelines. Thus, a 16S rDNA-Seq sparse count matrices simulator that considers the compositional nature of this data was developed. Then, a comprehensive benchmark analysis of forty-eight pre-processing pipelines was designed and performed to assess currently used and most-recent pre-processing approaches performance and to test for appropriateness in including zero-imputation step into 16S rDNA-Seq analysis framework.

Overall, this thesis considers the 16S rDNA-Seq data pre-processing problem and provide a useful guide for a robust data pre-processing when performing a 16S rDNA-Seq analysis. Additionally, the simulator proposed in this work could be a spur and valuable tool for researchers involved in developing and testing bioinformatic methods, thus helping in filling the lack of specific tools for 16S rDNA-Seq data.

Table of contents

Introduction

Thesis motivation

The study of microbial communities started from Leeuwenhoek's work in 1676 [1] and has changed incredibly during the following centuries. Microbes, that were initially ignored until the late 1800s, began to progressively occupy the centre of microbiologists' attention when their role in regulating and causing human diseases became evident. As a consequence, researchers began to develop a variety of techniques for isolating and growing microbes in the laboratory. Initially, microscopy was used to study their morphological features (rod, sphere, helix shape, ...), and culture techniques to classify microbes based on their nutrients and waste products. In the late 1970s, a breakthrough was brought by Carl Woese's discovery that ribosomal RNA (rRNA) genes could be used as molecular markers to perform organisms classification [2]. Some decades later, advances in molecular techniques allowed to access and describe microbial communities diversity in a culture-independent way. The advent of DNA sequencing technology indeed revolutionized the study of microbial communities, being the first mean of obtaining an overall view on a community content that released microbiological studies from the limit of observing only the small fraction of microbes growing in the laboratory. In parallel, the growing consciousness of the central role of the microbial population in regulating human and environmental equilibrium forced scientist in ever improving tools and method to investigate microbes diffusion, persistence and interconnections.

Today, one of the most widely used approaches for microbial communities investigations is based on the sequencing of the gene that codes for the small subunit of prokaryotic ribosome (16S rRNA gene). The ribosome is an essential piece of the cell's protein-making machinery, so 16S rRNA gene is present in all bacteria, but its exact DNA sequence is unique to each species. For this reason, it is used as a sort of molecular fingerprint for assigning to each community member a taxonomic characterization.

Population-wide microbial surveys became possible thanks to the advent of Next-Generation Sequencing (NGS) platforms, i.e. technologies that allow for deep, high-

throughput, in-parallel DNA sequencing. NGS instruments are able to produce a huge amount of data, enabling to sequence not only one microbial community but indeed a considerable number of different samples at dramatically reduced time and costs.

16S rRNA gene sequencing (16S rDNA-Seq) has become one of the most adopted methodologies in microbiome studies, due to its high amount of information at ever lowering time and cost expense. Exploiting the rapid progress achieved by NGS technologies, 16S rDNA-Seq allows large longitudinal, culture-free microbiome studies that permit a complete characterization of the studied niches.

This approach found its application in many determinant studies. For example, 16S rDNA-Seq permitted to demonstrate that diurnal oscillations in gut microbial localization and related metabolite production have a notable impact on the circadian epigenetic and transcriptional mechanisms of host tissues [3]. Another important application was proposed by in Peters et al. [4] in the framework of Public Health. In this study, 16 rDNA-Seq allowed to reconstruct a strong and consistent taxonomic signature of obesity.

In details, NGS instruments are able to produce millions of short sequences, called "reads", that are copies of original population genomic fragments. As for other sequencing approaches, 16S rDNA-Seq information is summarized into a matrix, where information on each sequenced sample content is reported. In particular, for 16S rDNA-Seq ("metataxonomic") studies each cell of this matrix contains the number of times ("counts") a read coming from a given sample was found to belong to a particular Operational Taxonomic Unit (OTU), an operational definition used to categorize bacteria based on sequence similarity. In practice, for each sample (column) the total number of reads obtained is split into "bacterial types" (rows), so that each column of the matrix (called "OTU table") represents the internal composition of the related microbial community.

OTU tables obtained from metataxonomic studies carry very peculiar characteristics that are linked to both biologic and instrumental reasons. First, the maximum obtainable sequencing depth constraint makes 16S rDNA-Seq data compositional, i.e. they are data in which each sample is a whole (composition) and the feature-wise counts are the components. The total number of reads imposed by the instruments implies each count being dependent from the level of presence of other taxa in the sample, making 16S rDNA-Seq data relevant only for proportional abundance considerations.

Secondly, 16S rDNA-Seq count matrices are typically highly sparse (70-95% of null values). This feature has two most probable causes. First, biological diversity in microbiome samples is usually abundantly present, meaning that most OTUs are characteristic of a specific subgroup of samples. Moreover, microbial population has a very skewed internal distribution, with a high number of rare or low-abundance species and a limited number of

highly present species. This fact, jointly with the finite number of reads obtainable from sequencing instruments, causes rare species loss at a rate that is heavily dependent on the internal microbiome distribution and on the number of samples sequenced in the same sequence run (the so-called multiplexing level).

The above peculiarities make bulk RNA sequencing tools and approaches unsuitable for 16S rDNA-Seq count matrices analyses. In particular, pre-processing steps, such as normalization, risk to introduce biases when a direct loan of RNA sequencing tools is done to perform data pre-analysis treatment on highly sparse matrices. In addition, current 16S rDNA-Seq count data work-flow does not consider a step in which species that became unobserved due to the sequencing step are treated for unobserved-data recovery (zero-imputation). This step is instead becoming a fundamental pre-analysis treatment in another sequencing framework, i.e. Single-cell RNA (scRNA-Seq) sequencing. scRNA-Seq data suffer from the same sparsity problem observed in 16S experiments and a huge scientific effort is now being made in order to find an appropriate zero-imputation strategy that could partially mitigate the information loss.

Despite being one of the preferential choices in microbiome studies, the continuous development of both experimental and computational procedures of 16S rDNA sequencing causes an unavoidable lack in standardization concerning sequencing output data treatment and analysis.

Thesis objectives

The research included in this thesis is motivated by the need of identifying the best computational methods for 16S rRNA gene sequencing (16S rDNA-Seq or 16S sequencing) data treatment, focusing on one of the most critical steps of sequencing data handling, i.e. 16S rDNA-Seq count data pre-processing.

Once the OTU (or "feature") table is obtained, a pre-processing step on the count data matrix is needed in order to make samples information comparable in spite of the possible different amount of reads obtained for each sample (sequencing depth) and to correctly perform the downstream analyses. This passage is called "normalization" and it is currently the only pre-processing step included in 16S rDNA-Seq data analysis.

Despite the considerable popularity of this technique, an exiguous number of specific tools are currently available for 16S rDNA-Seq data pre-processing that consider their peculiarities. Among these, the dramatic sparsity of count data matrices resulting from these studies is a major concern. In fact, due to instrumental technical limits and to typical internal microbial composition skewness (low number of high-abundant species and high

number of low abundant ones), 16S rDNA-Seq count data tend to loose information about rare species. As a consequence, many lowly abundant species that would be observed with no sequencing depth limitations result in zero counts. Most of the times, normalization is performed borrowing methods and tools from bulk RNA sequencing framework, where sparsity levels are heavily smaller. In relation to the amount of sparsity, this misuse may cause some biases in resulting normalized counts. Additionally, no pre-processing step is performed trying to recover abundances information that got lost during the sequencing process, contrarily of what it is now usually done in Single-Cell RNA sequencing framework, where also high sparsity levels are observed.

The main objective of this thesis was to identify optimal pipelines that filled the above gaps in order to assure solid and reliable conclusions from 16S rDNA-Seq ("metataxonomic") data analyses. To do this, three specific aims were identified: the study of the state-of-the-art pre-processing methods, the implementation of a 16S rRNA sequencing count data simulator and, finally, the benchmark of pre-processing pipelines for metataxonomic data.

The above main objective, in fact, inherently requires the use of in-silico datasets with known generating parameters and features that can be considered as the gold standard for comparisons. As a consequence, when working on identifying the optimal pre-processing pipeline for 16S count data it was of pivotal importance to first introduce a 16S sparse matrix simulator that was able to output realistic count data matrices on which testing the performance of pre-processing tools. The implementation of a new simulator was motivated by the lack of a tool specifically intended to directly obtain synthetic 16S rDNA-Seq count tables that properly models heavy sparsity and compositionality typical of these data. Indeed, currently the great majority of existing 16S sequencing simulating tools focus on simulating sequence libraries to deal with pre-OTU clustering steps and raw reads analysis [5]. These tools, such as *Grinder* [6], *GemSim* [7], *wgsim* (part of the *SAMTools* [8]) and *MetaSim* [9], permit the scientists to create their own mock community, but do not give direct access to an OTU table. Additionally, the few simulators available for direct count data construction, such as the ones implemented in R packages *PROPER* [10] or *ssizeRNA* [11] or the one proposed in Lee et al. [12], were mainly developed in bulk RNA sequencing context and they are consequently based on Negative Binomial count modeling, a framework that was demonstrated to be absolutely proper for RNA sequencing studies [13], but does not fit amplicon sequencing count data, due to their strong sparsity. Thus, the direct adoption of these tools for 16S sequencing simulations is not appropriate.

Thesis organization

In order to better contextualize the 16S rDNA-Seq scenario, in Chapter 2 some information on biological background, Next Generation Sequencing technology and microbiome studies are given. In Chapter 3, 16S sequencing count data are fully described, both in their obtaining procedure and in their constitutive characteristics. In the following Chapters, the 16S rDNA-Seq count data simulator developed in the context of this thesis, *metaSPARSim*, is presented, as well as the real data used to its performance assessment. In particular, in Chapter 4 two experiments that were part of this Ph.D. program and were performed to obtain real data for testing are described, jointly with one dataset available in literature that was also used for testing. In Chapter 5, a background on sequencing count data modelling as well as all the details about *metaSPARSim* characteristics and performances are introduced. In Chapter 6, normalization and zero-imputation steps are described, introducing both their theoretical framework and the state-of-the-art and widely used tools and approaches considered in the following for optimal pre-processing pipelines construction and testing. The metrics and methods used to evaluate pre-processing pipelines are introduced and explained in Chapter 7. In Chapter 8, results of the benchmark procedure performed on 48 different pre-processing pipelines are shown. Finally, strengths, limitations and future developments of the present work are discussed in Chapter 9.

Disclaimer

All the contributions described in this thesis, except where specific reference is made to the work of others, are the results of the research activity carried out within this Ph.D. program. The contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Chapter 1

Next Generation Sequencing and analysis of microbial communities

1.1 Biological background

1.1.1 DNA

Deoxyribonucleic acid (DNA) is an essential biomolecule present in all life-forms on Earth. It is composed of two chains made of nucleotides, building blocks of nucleic acids, that coil around each other to form a double helix carrying the genetic information used in the growth, development, functioning and reproduction of all known living organisms. Each nucleotide consists of three subunit molecules: a five-carbon sugar molecule, a nitrogenous nitrogenous (nitrogen-containing) base - which two together are called a nucleoside - and one phosphate group. In DNA, the sugar and phosphate group make up the backbone of the double helix, while the bases are located in the middle. A chemical bond between the phosphate group of one nucleotide and the sugar of a neighbouring nucleotide holds the structure together. Hydrogen bonds between the bases that are across from one another hold the two strands of the double helix together.

There are 5 bases that are commonly used in biochemistry and genetics: adenine, guanine, cytosine, thymine, and uracil, which have the symbols A, G, C, T, and U, respectively. The names of the bases are generally used as the names of the nucleotide, although this is technically incorrect. In fact, the bases themselves combine with the sugar to make the nucleotide adenosine, guanosine, cytidine, thymidine, and uridine. Nucleotides are named based on the number of phosphate residues they contain. For example, a nucleotide that has an adenine base and three phosphate residues would be named adenosine triphosphate (ATP). If the nucleotide has two phosphates, it would be adenosine diphosphate (ADP). If there

is a single phosphate, the nucleotide is adenosine monophosphate (AMP). Both DNA and RNA use 4 bases, but they don't use the same ones. DNA uses adenine, thymine, guanine, and cytosine, whereas ribonucleic acid (RNA) uses uracil instead of thymine. The helix of the molecules forms when two complementary bases form hydrogen bonds with each other. Adenine binds with thymine (A-T) in DNA and with uracil in RNA (A-U), while guanine and cytosine complement each other (G-C). Due to this fixed base pairing rule, the two strands are complementary since one DNA strand could univocally determine the sequence of its anti-parallel strand.

DNA can be roughly divided into two groups according to its function: genes and regulatory elements. Genes are regions of DNA which are transcribed by the enzyme RNA polymerase into RNA molecules. Regulatory elements, such as enhancers and promoters, are pieces of DNA where regulatory proteins called "transcription factors" can bind and then regulate the transcription of genes. Promoters are regions near the transcription start sites (TSS) of genes whereas enhancers are typically far away from the TSS. The probability of binding is linked to the shape of the protein and its matching with the particular sequence of DNA, as well as the presence of other proteins or molecules bound to DNA nearby.

The process which converts the information encoded in the DNA into the final product is called "gene expression" and starts by a first step, transcription, in which the genetic information stored in the DNA is transcribed into a RNA molecule.

1.1.2 RNA

The transcription of DNA into RNA is, in fact, the first step in the central dogma of molecular biology, the explanation of the flow of genetic information within a biological system. As stated by Crick in 1958 [14], "once information has passed into protein it cannot get out again. In more detail, the transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein may be possible, but transfer from protein to protein, or from protein to nucleic acid is impossible. Information means here the precise determination of sequence, either of bases in the nucleic acid or of amino acid residues in the protein.". Depending on their final product, the transcribed RNA molecules can be classified as:

- coding RNA: RNA molecules used to form proteins. The results of transcription is a messenger RNA (mRNA), which would be decoded into an aminoacid sequence by a ribosome through a process called "translation"
- non-coding RNA (ncRNA): RNA molecules which serve enzymatic roles as RNAs alone. They have control and regulatory functions and are involved in many cellular processes. Examples of ncRNAs are ribosomal RNA (rRNA) or transfer RNA (tRNA).

1.1.3 16S rRNA gene

The 16S ribosomal RNA (or 16S rRNA) is the component of the 30S small subunit of a prokaryotic ribosome that binds to the Shine-Dalgarno sequence, a ribosomal binding site in bacterial and archaeal messenger RNA that helps recruit the ribosome to the mRNA to start protein synthesis by aligning the ribosome with the starting codon. The genes coding for it are referred to as 16S rRNA gene (or 16S rDNA) and are present in almost all bacteria. This is because it has several important functions:

- the 3'end of 16S RNA binds to the proteins S1 and S21 known to be involved in initiation of protein synthesis;
- it has a structural role, acting as a scaffold defining the positions of the ribosomal proteins;
- it interacts with 23S, aiding in the binding of the two ribosomal subunits (50S+30S);
- it creates correct codon-anticodon pairing in the A site, via a hydrogen bond formation between the N1 atom of Adenine residues 1492 and 1493 and the 2'OH group of the mRNA backbone.

16S rRNA gene has a particular structure composed by the alternation of conserved and hypervariable regions. The former are highly conserved portions which enable PCR amplification of target sequences using universal primers, while the hypervariable regions, nine in total, demonstrate considerable sequence diversity among different bacterial species and can be used for species identification. A conceptual representation of 16S rRNA gene is reported in Figure 1.1.

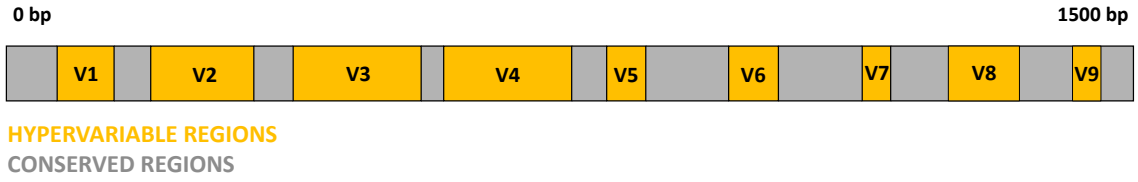


Fig. 1.1 16S rRNA gene structure.

1.2 Next Generation Sequencing

Next-generation sequencing (NGS) is a term used to identify the deep, high-throughput, in-parallel DNA sequencing technologies developed a few decades after the Sanger DNA sequencing method (see below). The aim of this section is to provide an overview of DNA sequencing methods history and of the major NGS platforms developed during the years, concluding with the description of currently most used sequencing approaches and instruments. Moreover, an introduction to 16S rRNA gene sequencing studies and details about the related experimental steps will be given.

1.2.1 NGS history

Since the definition of DNA structure [15], the way of investigating genomes properties and characteristics has changed dramatically. In this framework, a major role is played by DNA sequencing (DNA-Seq), i.e. the determination of the precise sequence of nucleotides that constitutes a DNA molecule. This approach allows for the precise characterization of a genome and the detection of possible variations between that genome and a reference one. These variations can include single nucleotide variants (SNV),inversions, deletions, insertions and region duplication. On the other hand, also RNA sequencing (RNA-Seq) has view a massive usage as a method for acquiring genomic information. In particular, it is widely used to identify mRNA transcripts, including novel transcripts and transcripts with alternative exons, and to measure the abundance of transcripts. Protocols developed to perform these two genome investigation approaches differ in some critical points. Firstly, the mRNA must be reverse transcribed into complementary DNA (cDNA) in order to be sequenced. In RNA-Seq, reads may come from both the template strand and coding strand of the gene ("unstranded" protocol), or using either the template or the coding strand ("strand-specific" protocol). Moreover, it is common in RNA-Seq to enrich for RNA molecules which end with a long string of adenosines (called "poly(A) tail") before the reverse transcription. This effectively increases mRNA molecules number over the highly abundant rRNA (ribosomal RNA) and tRNA (transfer RNA).

The first sequencing technique was developed in 1975 by Frederick Sanger [16], who used *Escherichia coli* DNA polymerase to rapidly copy single-stranded DNA molecules. Thanks to this technique, Sanger's team obtained a DNA sequence for the genome of bacteriophage ϕX174 of approximately 5,375 nucleotides length [17]. Despite the great sound of this result, this sequencing method had some important drawbacks; in fact, it was characterized by a low automation level and the information throughput was limited. This allowed to obtain copies of sequences of few hundreds nucleotides length at a time. The main progress in this context

was brought by selective incorporation of chain-terminating dideoxynucleotides (ddNTPs) in addition to typical deoxynucleotides (dNTPs) by DNA polymerase during in vitro DNA replication, again introduced by Sanger group [18]. This new method, known simply as the Sanger sequencing method, made DNA sequencing simpler, faster and more reliable and became the most widely used sequencing method for approximately 40 years. The first-generation automated DNA sequencers developed by Applied Biosystem Instruments (ABI), that used the Sanger method with fluorescent dye-terminator reagents for single-reaction sequencing rather than the usual four separate reactions, were later equipped with computers able to not only to collect and store sequencing data, but also to analyze them. The discovery of reverse transcriptase in 1970 [19, 20] led to the development of RNA sequencing using cDNA reverse transcribed from RNA. These advances, jointly with the creation of GenBank (http://www.ncbi.nlm.nih.gov/genbank) in 1982, permitted the generation and organization of a huge amount of DNA sequences throughout the 1980s, 1990s [21] and the publication of the first draft sequence of the human genome [22, 23] in the first years of 2000s. By the end of 2001, the complete genomic sequences of the bacteria *E. coli* and *Bacillus subtilis* the *Saccharomyces cerevisiae*, the nematode *C. elegans*, the plant *Arabidopsis thaliana* and the human genome were obtained. Sanger sequencing was by then the most widely used method, eve though sequencing was still time consuming and highly expensive. In 1996, the establishment of the first Affymetrix and GeneChip microarrays produced a rapid growth in DNA array technology and applications for various gene expression studies in prokaryotes and eukaryotes [24, 25]. However, DNA and RNA sequencing saw no decrease in use and popularity and new sequencing methods continued to rise to try to reduce the costs and hegemony of Sanger technologies [26]. These new methods were globally called "next-generation sequencing". They employed massively parallel strategies to output huge amounts of sequences from a large number of samples with high-throughput and with a high coverage level, thus allowing for high accuracy. These technological advances brought the cost of sequencing the genome down from $100 million in 2001 to less than $10,000 in 2014 [27].

The second-generation sequencing methods are then characterized by the need to prepare amplified sequencing libraries before performing the sequencing of the amplified DNA clones. The basic characteristics that mark second-generation sequencing technology are the following [21]:

- Shotgun sequencing of DNA or cDNA reverse transcribed from RNA is performed without the need for cloning via a foreign host cell; in fact, linker and/or adapter sequences are ligated to fragments for construction of template libraries.

- Library amplification is performed on a solid surface or on beads being isolated within emulsion droplets or arrays.
- Nucleotide incorporation is monitored directly by changes in electrical charge or by luminescence detection during the sequencing process.
- NGS generates millions of nucleotide reads in parallel in a much restricted time than by the Sanger sequencing method. Additionally, NGS reads are digital and therefore enable direct quantitative comparisons.
- Either single or pair end reads can be obtained at fragment ends.

On the contrary, the great innovation of third-generation single molecular sequencing is that it can be done without the need for creating the time-consuming and costly amplification libraries. NGS technologies have been reviewed in several works [21, 28–31]; we restrict here our interest to the most-used second- and third-generation platforms for 16S rDNA-Seq, that are reported in the following section as presented in [21].

1.2.2 NGS platforms

Second-generation platforms

Roche 454 pyrosequencing

Roche 454 pyrosequencing by synthesis (SBS) was the first commercially successful second-generation sequencing system. It was developed by 454 Life Sciences in 2005 and later acquired by Roche in 2007. In this technology, visible light is detected and measured after it is produced by an ATP sulfurylase, luciferase, DNA polymerase enzymatic system in proportion to the amount of pyrophosphate that is released during repeated nucleotide incorporation into the newly synthesized DNA chain. The system can produce more than 200,000 reads of 100-150 bp per read, with a consequent output of 20 Mb per run in 2005 [32]. In 2008, Roche released the 454 GS FLX Titanium system, that improved the average read length to 700 bp with an accuracy of 99.997%. The major drawbacks of this technology are the high cost of reagents and high error rates in homopolymer repeats. Additionally, supply or service to the 454 sequencing machines or the pyrosequencing reagents and chemicals were stopped in 2016.

Illumina (Solexa) sequencing

Illumina purchased the Solexa Genome Analyzer in 2006 and commercialized it in 2007 [33]. Nowadays, it dominates sequencing business (>70% dominance of the market), particularly with the HiSeq and MiSeq platforms. The Illumina sequencer adopted the sequencing by synthesis, using removable fluorescently labeled chain-terminating nucleotides

that are able to produce a larger output at lower reagent cost [31]. The template DNA for sequencing is generated by PCR bridge amplification (also called "cluster generation"). The output of sequencing data per run is higher, the read lengths are shorter, the cost is cheaper, and the run times are much longer than most other systems [21]. Illumina provides seven sequencing machines (iSeq, MiniSeq, MiSeq, NextSeq, HiSeq, HiSeq X and NovaSeq) with mid to very high output (1.2 Gb - 6 Tb). The MiSeq, which, although small in size, has a maximum output of 15 Gb and fast turnover rates suitable for targeted sequencing. Illumina's new method of synthetic long reads using TruSeq technology apparently improves de novo assembly and resolves complex, highly repetitive transposable elements [34].

Sequencing by Oligonucleotide Ligation and Detection (SOLiD)

Supported Oligonucleotide Ligation and Detection (SOLiD) is a next-generation sequencer introduced by Life Technologies and first released in 2008 by Applied Biosystems Instruments (ABI). It is based on 2-nucleotide sequencing by ligation (SBL) [31]. This procedure is made by sequential annealing of probes to the template and their subsequent ligation. Sequencers on the market today are suitable for both small- and large-scale studies involving whole genomes, exomes, and transcriptomes and have enabled greater throughput and simpler workflows by replacing previously used beads with direct in situ amplification on FlowChips and paired-end sequencing. The advantage of this technology is accuracy, while the major disadvantages are the short read lengths, the very long run times (several days), and the need for state-of-the-art computational infrastructure and expert computing professionals for raw data analysis.

Ion torrent

Ion Torrent technology was developed by the same inventors of 454 sequencing [35], that introduced two major changes. First, the nucleotide sequences are read electronically by changes in the pH of the surrounding solution in relation to the number of incorporated nucleotides rather than by the generation and detection of light. Second, the sequencing reaction is performed within a microchip that is amalgamated with flow cells and electronic sensors at the bottom of each cell [21]. The incorporated nucleotide is then converted to an electronic signal detected by the sensors. The most used sequencers in the market that use Ion Torrent technology are the Proton sequencer, the Ion Personal Genome Machine (PGM), and the recently released GeneStudio S5. There are five sequencing chips to choose from, that allow for both high- and mid-throughput, following the study needs . Sample preparation for the generation of DNA libraries is simplified by the useful Ion Chef system available for automated template preparation and chip loading. The Ion Torrent chip is used with an ion-sensitive field-effect transistor sensor that has been engineered to detect individual protons produced during the sequencing reaction. The chip is placed within the flow cell

and is sequentially flushed with individual unlabeled dNTPs in the presence of the DNA polymerase [21]. The incorporation of a nucleotide into the DNA chain releases H protons and consequently changes the pH of the surrounding solution in a way that is proportional to the number of incorporated nucleotides. The major disadvantages of the system are problems in reading homopolymer sequences and repeats. The major advantages are the relatively longer read lengths, a flexible and partially automated workflow, reduced time, and a cheap price.

Third-generation platforms

Single-molecule real-time (SMRT) sequencing by pacific biosciences

Pacific Biosciences markets the PacBio RS II sequencer and the SMRT real-time sequencing system. SMRT sequencing is performed in Single-molecule real-time cells that contain 150,000 ultra-microwells at a zeptoliter scale where a single DNA polymerase molecule is captured at the bottom of each well using the biotin-streptavidin system in nanostructures known as zero-mode waveguides (ZMWs). Once the single-strand template DNA is coupled with immobilized DNA polymerase, fluorescently labeled dNTP are added and detected when the nucleotide is incorporated into the copy strand. charge-coupled device (CCD) cameras continuously detects the ZMWs that are converted into single molecular traces representing the template sequences. Since all four nucleotides are added simultaneously, the speed of sequencing is much increased compared to technologies where individual nucleotides are inserted sequentially. The initial reported accuracy was 99.3% with read lengths of about 900 bp [31]; the template circularization and repeated sequencing using a technology called SMRTbell templates provided longer reads and improved the accuracy to >99.999% [36]. Once sequencing is initiated, the Blade Center computational system performs real-time signal processing, base calling, and quality assessment. Primary analysis data, including read-length distribution, polymerase speed, and quality measurement is passed directly to the secondary analysis software called SMRT Analysis that is capable of processing sequencing data in real time. This also includes a full suite of tools to analyze sequencing data.

Nanopore sequencing by Oxford Nanopore Technologies (MinION and PromethION)

Oxford Nanopore Technologies provides the latest single-molecule sequencing system [37]. The MinION Mkl is a portable device for DNA and RNA sequencing that can be directly attached to a laptop/computer using a USB port, while the PromethION is a small bench-top system. The idea at the basis of DNA and RNA sequencing using nanopores is that the conductivity of ion currents in the pore changes when the strand of nucleic acid passes through it. The flow of ion current depends on the shape of the molecule passing through the pore and, since nucleotides have different shapes, each nucleotide is recognized by its effect on the change of the ionic current [38]. The key advantage of this approach is that

sample preparation is minimal compared to second-generation sequencing methods, and long read lengths can be obtained in the kilo-bp range. In addition, there are no amplification or ligation steps required before sequencing. The main drawback is the requirement to optimize the speed of DNA passage through the nanopore to ensure reliable measurement of the ionic current changes and reduce the high error rates of base calling [37, 38].

1.3 Microbiome studies

"Microbiome" is the term used to identify all of the genetic material within a *microbiota*, i.e. the entire collection of microorganisms in a specific niche. Thus, with this term we include the whole microbial community of commensal, symbiotic and pathogenic microorganisms present into an precise ecosystem. In this section, a brief history of microbial community studies is presented, as well as a delineation of microbiome studies framework, aims and tools.

1.3.1 Brief history of microbial community studies

In Begon et al. [39] microbial communities are defined as sets of organisms (in this case, microorganisms) coexisting in the same space and time. The study of microbial communities has strongly changed from the first discoveries about microbes made by Leeuwenhoek in 1676 ([1]) to the current characterization using molecular techniques. First efforts were made by scientists with the aim to isolate these invisible organisms, starting from solid nutrients like gelatine or potato slices to cultivate, isolate, count and visualize them. These embryonic steps helped microbiologists to understand microorganisms physiologies and the rapid improvement of the resolution of microscopy techniques brought real burst in the study of microorganisms and their interaction. For almost 300 years, the study of microorganisms was based on morphology features, growth, and selection of some biochemical profiles ([40]). These techniques provided an insight into the microbial world, but nowadays, they provide only a limited resolution for other applications.

In the late 1970s, a notable breakthrough was introduced by Carl Woese, who proposed the use of ribosomal RNA (rRNA) genes as molecular markers for organisms classification [2]. His idea, jointly with the above mentioned Sanger automated sequencing [18] method, revolutionized the way microorganisms were studied and classified. Some decades later, advances in molecular techniques allowed to access and describe microbial communities diversity in a culture-independent way. In fact, PCR, rRNA genes cloning and sequencing, fluorescent in situ hybridization (FISH), denaturing gradient gel electrophoresis (DGGE and

TGGE), restriction-fragment length polymorphism, and terminal restriction-fragment length polymorphism (T-RFLP) brought valuable advantages to this field.

In spite all these improvements, microorganisms metabolic and ecological functions remained opened questions in microbiology. The investigation in this framework was possible only after gene cloning from total DNA. This led to the development of gene expression techniques, that allowed to discover new molecules and to identify new microbial communities members.

1.3.2 Reconstructing microbial community content from NGS data: metagenomics and metataxonomics

Currently, there are two main strategies to perform the analysis of microbial communities through NGS: shotgun genomic sequencing ("metagenomics") and amplicon sequencing ("metataxonomics" [41]). In shotgun metagenomics, bacterial DNA is isolated from the whole microbial community and sequenced. The resulting reads are then analyzed using metagenomics databases as a reference to obtain a taxonomical assignment for each sequencing read. Conversely, amplicon sequencing relies on PCR amplification of specific target genes, the most used of which in bacterial community studies is the 16S rRNA gene. So-obtained PCR amplicons are then sequenced.

16S sequencing is a robust, well-characterized method that permits to obtain sufficient information about microbial communities composition, starting from a relatively small number of sequences per samples [42] and allowing for the sequencing of a high number of samples at a time. However, a major limitation of this method is that taxonomy is reconstructed on the basis of the sequence of only a single region of the bacterial genome. This causes the choice of primers used to amplify rDNA to be a crucial and critical step in experimental design, as some primers have been shown to exhibit a bias resulting in over- or under-representation of specific taxa [43].

On the other hand, shotgun metagenomics requires higher coverage (10–30 million of reads [42]) and a more complex downstream data analysis. Nevertheless, by collecting sequence information about broad genomic regions, shotgun metagenomics allows a more accurate definition at the species level, thus yielding a detailed description of bacterial community. However, a recent work by Cloney et al. [44] demonstrated that shotgun metagenomics and 16S rDNA sequencing describe the bacterial composition yielding comparable results.

In the following, we focus on 16S microbiome studies, as the framework of interest of this thesis, and we show in detail all the steps of a typical 16S rDNA sequencing experiment.

1.3.3 Typical 16S microbiome experiment

Metataxonomics using 16S sequencing is a widely used technique that relies on the alternation of conserved and hypervariable regions of the bacterial 16S rRNA gene to make community-wide taxonomic classifications. As above introduced, 16S rRNA gene plays an essential role in bacteria life; as a consequence, they are very highly conserved in time, that means it is possible to construct a phylogeny (tree of life) linking together all known bacteria. Additionally, it consists of both conserved and hypervariable regions, so that one can design a universal primer for the conserved regions to target all the bacteria. By running PCR, one can indeed amplify the 16S rRNA gene covering chosen hypervariable regions and determine the differences in the obtained sequences to detect the presence and the abundance of various species by counting how many obtained 16S rDNA sequences belonged to each particular species. Additionally, the information about the degree of difference in the bases of hypervariable region can be used to determine their phylogenetic closeness. Metataxonomics has been widely used due to its convenience to perform taxonomic and phylogenetic classification in large and complex samples within organisms from different life domains.

In this section, the basic workflow for 16S rRNA gene sequencing is described, starting from sample collection to sequencing process, following the sequence of its main steps.

1.3.3.1 Experimental design: 16S rRNA or rDNA?

When performing a microbial community study, a main question researchers have to pose themselves regards which aspect of the microbiota they are interested to, the mere community composition or its metabolically active part. Depending on the answer, one can in fact decide to perform the sequencing on 16S rDNA or on reverse-transcribed rRNA. Metabolically active bacterial cells are usually characterized by a higher amount of ribosomes than resting or dormant cells [45]. Therefore, sequences obtained from reverse-transcribed rRNA are better indicators of the active bacterial populations at the time of sampling than sequences from rDNA templates.

This initial choice is relevant in this context because, in case rRNA sequencing is performed, an additional step in the experimental workflow, that could potentially add supplementary bias in final data, has to be considered. In fact, current sequencing technologies work with DNA sequences as input, so the rRNA must be converted to double stranded complementary DNA (cDNA) prior to sequencing step.

1.3.3.2 Samples processing and library preparation

The first step in a 16S rRNA/rDNA sequencing experiment is isolating and purifying RNA. First, cells are disrupted using detergents, chaotropic agents and, depending on the sample/experimental protocol, mechanical methods. RNA/DNA is then extracted from the total cell lysate through the use of organic solvents or solid-phase extraction onto silica. Finally, a quality and quantity check is performed to assess material preservation level and separation of RNA/DNA from cellular materials.

As already introduced, if the starting material was made by RNA an additional step has to be performed. This is justified by many reasons.

First, RNAses are ubiquitous in nature, so RNA is typically unstable. In contrast, DNA highly more biologically stable, so converting RNA to DNA ensures the stability of the sample information content. Additionally, DNAses can easily be inactivated by chelating their metal ion cofactors, while RNAses do not require metal ion cofactors and are therefore much harder to inactivate so that obtaining a sample free of RNAses can be quite difficult.

Second, PCR amplification only works on DNA. Although it would be possible to adapt PCR to RNA using RNA-dependent RNA polymerase instead of DNA polymerase, RNA-based PCR protocol would introduce far more errors than the combination of reverse-transcription and DNA-based PCR.

For the above reasons, RNA must be converted to double stranded complementary DNA (cDNA). To do this, total RNA is incubated with:

- a particular type of polymerase, the reverse transcriptase (RT)
- deoxyribonucleotide triphosphate (dNTPs)
- bivalent Mg^{2+} ions, which act as cofactors for the polymerase
- a buffer solution
- primer sequences, that act like starters, pairing in a complementary way to the RNA filament

The most used primers are *oligo-dT* sequences and *random primers*. oligo-dT are oligonucleotide sequences of thymidine designed to match the polyadenylated RNA tail. This technique works only for polyadenylated RNA, so to amplify a sample of RNA without the polyadenosine tail, the most suitable primers are the so-called "random hexamers" or i "random nonamers". These are random sequences of 6 or 9 bases which, with their randomness, can potentially act as primers for any RNA template sequence. Some researchers use a combination of both primers to get a mixture with the characteristics of both.

The primer provides a free 3'-OH usable by reverse transcriptase to generate a DNA strand complementary to the transcript. This phase is called elongation and its course is strictly dependent on the nature of the enzyme. In fact, also for enzyme choice there are different options. When using the enzyme MMLV (Moloney murine leukemia Virus), the elongation reaction should be carried out at 37°C, which is problematic in the case of a rich RNA in secondary structures and/or GC. A better choice is the enzyme Superscript III (SSIII), a variant of the enzyme MMLV engineered to withstand higher temperatures. In this case, the reaction is carried out at 50°C for about 1 hour. The SSIII enzyme also has a reduced H RNase activity compared to MMLV, which results in a longer cDNA and better yield.

When the first strand is synthesized, before proceeding with the amplification of cDNA, the RNAse enzyme (usually derived from the bacterium *Escherichia coli*) is added to the reaction, degrading the original RNA filament that has been used as a reverse transcriptase template. When retrotranscription is complete, the generated cDNA is amplified using a standard PCR method. A thermostable DNA-dependent DNA polymerase, an enzyme with a 5'-3' polymeric activity, is added to the cDNA template and in the presence of a pair of specific primers for the 16S rRNA gene sequence to be amplified, the PCR reaction is initiated.

First, a double-stranded DNA molecule is synthesized from the cDNA molecule, leading the reaction to a suitable temperature to allow priming of the DNA primers. Subsequently, bringing the temperature to 95°C, the new DNA molecule is denatured and the two separate filaments are ready for a new primer pairing and for the polymer synthesis, at a new temperature drop (about 72°C). After approximately 30 cycles, millions of copies of the sequence of interest will are produced. When starting from DNA, this last step is directly performed.

The final product of the previous step is the so-called *library*, i.e. the collection of DNA fragments that is stored in a the studied population.

1.3.3.3 Adaptor inclusion and indexing

The next step in library preparation consists in fragmenting DNA and adding adapter sequences at the ends of the fragments. Adapters are required both for clonal amplification and for priming the sequencing reaction. Adapters composition is specific for the each sequencing platform and, consequently, they contain several optional elements employed by different techniques (e.g. multiplexing, paired-end sequencing,...). For instance, sequence indexing, i.e. the inclusion in adapters of specific indices (or barcodes), allows to uniquely identify the provenance of each sequence when performing multiplexing. This procedure allows to pool different libraries in a single sequencing reaction, saving both time and money.

Another useful procedure involves adapters containing sequencing priming sites for the opposite sides of the fragment. These adapter elements allow sequencing of both ends of the fragment (paired-end sequencing), resulting in a higher coverage and a more precise information.

The final step in library preparation consists of a number of PCR cycles to enrich for product that has adapters ligated to both ends. As amplification is known to be the source of several biases [46], it is advisable to minimize the amplification steps are often reduced to the bare minimum.

1.3.3.4 Sequencing

Sequencing is the last step in the data production process. As seen in Section 1.2, numerous sequencing platforms available to perform this task, each one characterized by different proprietary technologies and chemistries. The current leading platform is Illumina, followed by IonTorrent and PacBio. Corresponding to different features, each sequencing platform has unique strengths and drawbacks; moreover, the choice of platform depends on the goal of the study. The appropriate sequencing platform should be also selected based upon the sequencing coverage and its relation to the number of samples to be run. When studying core members (present in a high percentage of samples) of a microbial community, a good strategy to decrease costs may consist in lowering the coverage by increasing the number of samples in a sequencing run. However, if rarer members of a community are of interest lower sample numbers leading to increase coverage may be more appropriate. The choice of the sequencing platform should then also consider the desired throughput to address the study aims and time/cost constraints.

Chapter 2

16S sequencing count data

The output of a NGS process is composed by a huge amount of raw sequences, i.e. reads, divided into one ore more files. In order to extract useful information from sequencing data to profile the studied community, these sequences have to be appropriately treated to perform downstream analyses, following a complex analysis pipeline (Figure 2.1). In the first part of the pipeline (Read processing), reads obtained from sequencing are analyzed with the aim of identifying their provenance, i.e. to which organism the sequence belongs to. Once the raw sequencing data are organized in the so-called count matrices, the second part of the pipeline (Count data processing) exploits the collected information to perform a wide range of downstream analyses. Prior to these analyses, a fundamental step, the count data pre-processing, is performed to mitigate the biases present in data due to the sequencing process. The detailed description about read and count data processing steps is provided in the next sections.

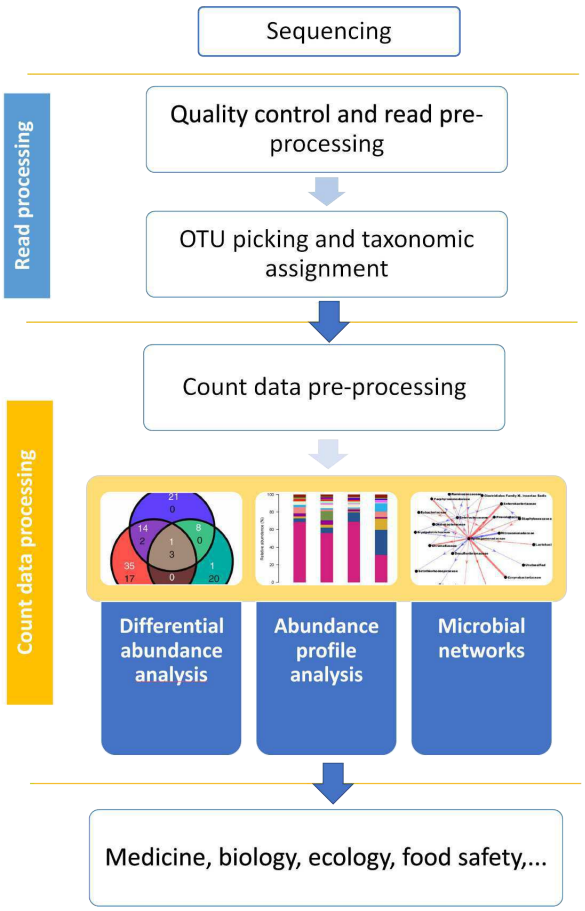


Fig. 2.1 16S rDNA-Seq analysis pipeline. Flowchart of a typical 16S rDNA-Seq analysis pipeline. The first part of the pipeline (Read processing) handles sequencing reads. The second part (Count data processing), exploits the 16S rDNA-Seq data information resumed in a count table to perform a variety of downstream analyses.

2.1 From reads to counts: the importance of being earnest

The ultimate goal of this first part is to bin in clusters the sequenced reads according to their sequence similarity to each other, resulting in cluster of reads, commonly referred to as Operational Taxonomic Units (OTUs), and to finally resume sequencing information into the so-called OTU table, a matrix of count data representing the number of reads obtained for each OTU in each sample. Eventually, in the ideal scenario each OTU should represent an actual bacterial species. This result is non straightforward and has to deal with possible biases introduced during the sequencing procedure. In this section, each pre-processing step needed to go from raw reads to the final OTU table is considered and analyzed, highlighting its importance, possible pitfalls and drawbacks of wrong pre-processing decisions.

2.1.1 Primer removal and demultiplexing

Once reads are obtained from the sequencer, the first step consists in demultiplexing data, separating reads coming from each sample present in the starting library mix. After that, barcodes (or indices) used for sample multiplexing have to be removed, as well as primer sequences, as these interfere with the taxonomic analysis. Reads not matching any barcode are discarded. If the experiment was performed sequencing both ends of fragments (paired-end sequencing), forward and reverse reads are then joined together.

2.1.2 Sequence quality trimming

Raw reads generated by a next-generation sequencing machines generally are equipped by predicted error probabilities for each base call indicated by quality scores. In many applications it is important to filter reads to reduce the number of errors, especially in marker gene sequencing experiments such as 16S rDNA-Seq. The next step in the pipeline is then aimed at discarding reads or part of them with insufficient quality scores. This could be done in several different ways. For example, reads can be selected imposing that all bases should have a minimum quality score or discarding a portion of read (head or tail) for which a drop in quality from a base on is detected. One of the most used approach uses a sliding window to possibly trim the read starting from a point in which the mean quality over n considered bases is under a specified threshold. As the chosen method will affect final results, in this step it is crucial to chose appropriately the most-fitting one for the analyzed data, in order to find the right trade-off between retaining good quality reads and not discarding too much information.

2.1.3 Chimera checking and read denoising

An important source of errors that has to be considered in this stage originates from chimera formation within the PCR amplification step, that is the possible formation of a chimeric sequence which consists of two or more fragments from distinct species [47]. As these chimeras will undergo the same processes as any other DNA sequence, they can take a large portion of all unique sequencing reads and have to be removed to avoid misleading information. Choices made in removal of these artificial sequences will have a huge impact on the community diversity estimates, i.e. in the numerosity of detected OTUs, since chimeras that go undetected will be interpreted as novel species.

After chimera detection and removal, a dereplication step is also typically applied at this stage, to combine all identical sequencing reads into “unique sequences” with a correspond-

ing “abundance” equal to the number of reads with that unique sequence. Dereplication substantially reduces computation time by eliminating redundant comparisons.

Then, a final denoising of data is performed. Although there is a great variety of programs available for error removal in sequencing read data, which differ in the error models and statistical techniques they use, the parameters, data structures and algorithms they use, the final aim is always to obtain data as free as possible from errors.

2.1.4 Read clustering and taxonomic assignment

This is the read-processing step that concludes the first part of the typical 16S rDNA-Seq analysis pipeline. In this stage, the information contained in pre-treated reads is organized into a unique table in which the microbial composition of sequenced sample is quantitatively described. Two approaches are commonly used to achieve this goal: taxonomy-dependent methods and OTU-based methods [48–51]. The taxonomy-dependent methods are based on the query of sequences against databases of already annotated for taxonomic assignment. On the contrary, in the OTU-based methods all the sequences are clustered into Operational Taxonomic Units (OTUs) based on a distance matrix at a specified threshold. The lack of sufficiently well-characterized microbes and reliable taxonomy often makes it difficult to characterize novel sequences using taxonomy-dependent methods, and their robustness and accuracy are mainly dependent on the completeness of the annotated reference database. Another limitation is that most existing reference databases are deeply-characterized only from the genus level up, rather than at the species level. In contrast, OTU-based methods allow to assign all sequences into OTUs without prior information, so that all sequences can be processed, including both microbes that have not been annotated in the databases as well as novel uncultured ones. For these reasons, OTU-based methods are the preferred way to summarize 16S rDNA sequencing experiments information. Clearly, also the OTU-based methods have some issues that need to be addressed for their successful applications, such as the presence of sequencing errors which would result in an over-estimation of really present OTUs or the heterogeneous evolution rates in 16S rDNA which make it difficult to define a consistent threshold to separate OTUs.

In general, the OTU-based methods can be categorized into hierarchical clustering, heuristic clustering and model-based clustering methods [52].

In the first case, the difference between each pair of sequences is measured by a distance matrix, and standard hierarchical clustering is then used to define OTUs at a specific level of sequence dissimilarity. The main issue liked with these methods is the computational complexity, that cause this approaches not being the most suitable solution when dealing with large-scale sequencing data.

In heuristic clustering, for a fixed threshold an input sequence is selected as a seed for the initial cluster and then each input sequence is examined sequentially. If the distance between the query sequence and representative sequences of the existing clusters is under the pre-fixed threshold, the input sequence is be added to the corresponding cluster, otherwise a new cluster is created and the query sequence is stored as a new seed. Heuristic clustering algorithms have a lower complexity at the cost of reduced biological accuracy; that is, there is a trade-off between complexity and accuracy.

Finally, to avoid using for a hard threshold definition, as required by hierarchical and heuristic methods, a Gaussian mixture model-based clustering algorithm called Clustering 16S rRNA for OTU Prediction (CROP) [53] was proposed. It is based on an unsupervised probabilistic Bayesian clustering algorithm and uses a soft threshold for defining OTUs. The CROP algorithm avoids setting an often subjective and critical-to-find threshold, thus possibly reducing the effects of PCR and sequencing errors in inferring OTUs.

Recently, new methodologies have been developed that obtain amplicon sequence variants (ASVs) from Illumina amplicon data without the need to specify the arbitrary thresholds that differentiate OTUs [54]. ASVs are single-nucleotide differing biological sequences inferred in the sample prior to the introduction of amplification and sequencing errors. This is obtained by a *de novo* procedure in which sequence variants are discriminated from errors considering that biological sequences are more likely to be observed multiple times than are error-containing sequences. Inherently from their construction process, ASVs can capture all biological variation present in the samples and can be reproduced in other datasets and validly compared between different datasets. The most popular tool in this context is DADA2 pipeline [55], whose output is a higher-resolution analogue of an OTU table in which the number of times each exact amplicon sequence variant was observed in each sample is reported. It is noteworthy that, due to the increased resolution, these count matrices are typically characterized by a heavier sparsity level than the one observed classical OTU tables.

When taxonomy-dependent methods are chosen, the taxonomic description of each cluster is automatically obtained during the reads binning. On the contrary, OTU-based methods require this passage to be done after cluster formation. In this case, a reference sequence is elected for each OTU and the taxonomic profiling is obtained looking for the best-matching reference sequence into available databases [56–58].

The final output of the first part of the pipeline is then a count matrix, often referred to as OTU table, in which for each sample (matrix columns) the number ("count") of reads coming from each OTU (matrix rows) is reported, as exemplified in Table 2.1.4. This way of representing data information as counts is commonly adopted for resuming data coming for different sequencing approaches, such as, for example, RNA sequencing, in which OTUs are

substituted by genes and counts represent the expression level of each gene in each sample. Despite the existing analogy in formal representation, count matrices coming from different sequencing approaches have basic characteristics that are extremely peculiar to the performed experiment and that have been appropriately described by different underlying models. For instance, as explained in Chapter 5, the Negative Binomial distribution is usually chosen to model RNA-Seq count data matrices, while for 16S rDNA-Seq data matrices this model is not the most appropriate choice due to the higher sparsity and variability (see Chapter 5).

Table 2.1 Count table structure

	Sample 1	**Sample 2**	**Sample 3**	**...**	**Sample P**
OTU 1	$C_{1,1}$	$C_{1,2}$	$C_{1,3}$	...	$C_{1,P}$
OTU 2	$C_{2,1}$	$C_{2,2}$	$C_{2,3}$	...	$C_{2,P}$
OTU 3	$C_{3,1}$	$C_{3,2}$	$C_{3,3}$	...	$C_{3,P}$
...	...	...	...	...	...
OTU M	$C_{M,1}$	$C_{M,2}$	$C_{M,3}$	...	$C_{M,P}$

In the next section, the main characteristics of 16S rDNA-Seq count matrices are introduced, in order delineate the precise framework in which this thesis is included, that is the proper pre-processing treatment of 16S rDNA-Seq count data. This constitutes the first step included in the second portion of the pipeline in Figure 2.1, to which the following Chapter 3 is entirely dedicated.

Decisions taken when performing pre-processing operations on count data are critical in the sense that they will greatly influence all downstream analyses subsequently performed and, of course, final results and conclusions [59]. For this reason, it is of preeminent importance to deeply understand and describe the peculiarities of analyzed data, in order to choose the most appropriate correction before performing any type of downstream analysis.

2.2 16S rDNA sequencing count tables

OTU count tables coming from 16S microbiome studies have three major characteristics: they are non-negative, over-dispersed, and have a huge number of zeros [60].

The positivity comes directly from the fact that OTU tables are count tables, i.e. matrices in which entries are counts of elements within a particular class (row) and sample (column).

The term "overdispersion" is used in statistics to indicate the presence of greater variability (statistical dispersion) in a data set than would be expected based on a given statistical model. As explained in Chapter 5, the most-used models for count data are Poisson and Negative Binomial models. As recently explained by Xu et al. [60], 16S rDNA sequencing data

typically exhibit a constitutive over-dispersion, with this definition meaning that Poisson or Negative Binomial models, even though very used for general count or bulk RMA-seq data, are not appropriate to catch and describe all the variability present in 16S rDNA-Seq data.

Another important characteristic of these data is the high percentage of zero counts. This phenomenon, called zero inflation, has two main sources, a biological and a technical one.

The first contribution to sparsity is primarily due to the fact that the OTUs are subject dependent, i.e. their presence and abundance are unique in each subject. As a consequence, only a few major bacterial taxa of the microbiota are shared across samples, whereas the remaining part is detected only in a small amount of the samples. In this case, zero counts in the sample are simply justifies by the fact that the corresponding OTUs are truly absent. These null values are commonly referred to as biological or structural zeros

On the contrary, zero counts may also be linked to rare OTUs, i.e. with low abundance, that were indeed present in the sample but were not observed because of insufficient sequencing depth. This could be due to several causes, such as a bad performance sequencing experiment or an excessive multiplexing level, and leads to the formation of the so-called technical or sampling zeros.

2.2.1 Simple(x), it's compositional! Why we should treat 16S rDNA-Seq data as compositions

A separate section is here dedicated to another very important 16S rDNA-Seq data aspect: compositionality. In fact, as recently recalled by Gloor et al. [61], data obtained from high-throughput sequencing (HTS) of 16S rRNA gene amplicons are compositional because they have an arbitrary total, the sequencing depth, imposed by the instrument and the read count observed in a HTS run is a random sample of the relative abundance of the molecules in the original sample. Moreover, counts cannot be linked to the absolute number of molecules in the input sample, but are naturally described as proportions or probabilities (sum to 1), or with a constant or irrelevant sum. Data with these characteristics are referred to as compositional data. These data are constrained by the simplex and are not free floating in the Euclidean space; therefore, standard methods of analysis are not applicable, because a dependency structure between OTUs (also called "parts" in compositional alnalysis framework) is present. For example, an increase in abundance of one prevalent bacterial taxon can lead to spurious negative correlations for the abundance of other taxa.

Compositional data definition was introduced by Aitchison in 1986 [62]. In its work, Aitchison pointed out some main features of compositional data that have to be taken into

account when dealing with them. The main principles underlying compositional analysis are [63]:

- Scale invariance: vectors with proportional positive components represent the same composition. In other words, if a composition is scaled by a constant, e.g. changing from percentages to parts per unit, the information carried is completely equivalent
- Subcompositional coherence: analyses concerning a subset of parts must not depend on other non-involved parts.

On the basis of its observation, Aitchison built a new geometry, called Aitchison geometry of the simplex, in which, given $x, y \in S^D$ and $\alpha \in \mathbb{R}$, the simplex S^D has a Euclidean space structure with:

- **perturbation:** $x \oplus y = C[x_1 y_1, \ldots, x_D y_D]$, with C being the closure operation;
- **powering:** $\alpha \odot x = C[x_1^\alpha, \ldots, x_D^\alpha]$;
- **inner product:** $\langle x, y \rangle_\mathrm{a} = \frac{1}{D} \sum_{i=1}^{D-1} \sum_{j=i+1}^{D} \left(ln \frac{x_i}{x_j} ln \frac{y_i}{y_j} \right)$
- **norm:** $||x||_a^2 = \frac{1}{D} \sum_{i=1}^{D-1} \sum_{j=i+1}^{D} \left(ln \frac{x_i}{x_j} \right)^2$
- **distance:** $d_A(x, y) = \sqrt{\frac{1}{D} \sum_{i=1}^{D-1} \sum_{j=i+1}^{D} \left(ln \frac{x_i}{x_j} - ln \frac{y_i}{y_j} \right)}$

The whole structure allows to construct a basis on the simplex so that standard statistical methods designed for the Euclidean space can be applied. Out of several proposals for the construction of a basis [64], the isometric logratio (ilr) transformation [65] seems to be the most convenient one. The ilr transformation offers good theoretical and practical properties [65]. One important property is the isometry, meaning that the Aitchison distance of two vectors is the same as the ordinary Euclidean distance for their ilr images. Thus, the ilr transformation allows to represent compositional data in terms of the standard Euclidean geometry, and therefore standard statistical methods can be applied. Note that this property is also fulfilled for the centered logratio (clr) transformation originally proposed by Aitchison [62], but this transformation results in singular data, causing problems for robust estimation.

Chapter 3

Count data pre-processing

As explained in Chapter 2, microbial community sequencing data are typically summarized into large matrices where the columns represent samples and the rows contain OTU count values, that represent bacteria types. As pointed out in [59], several characteristics of OTU tables can cause incorrect results in downstream analyses if ignored. First, the total microbial community in each biological sample may be represented by very different amount of sequences (i.e., library sizes), sometimes by several orders of magnitude, reflecting differential efficiency of the sequencing process rather than real biological variation. This fact is confirmed by the observation in most of the studies that the number of detected species does not correspond the totality of species present in the sample. In fact, rarefaction curves often reveal that the species number is rarely saturated, implying that more bacterial species would be observed with more sequencing depth. Thus, some of the most used metrics such ad alpha and beta diversity could carry erroneous information, because some OTUs may be scored as unique to certain samples only due to their superior sequencing depth. Second, the typical high sparsity (70-90%) of these count tables implies that the counts of rare OTUs are uncertain, since they are near the limit of detection. Finally, as seen in Section 2.2.1, OTU abundances have a compositional nature ([66], [67], [68]) that is not usually taken into account. In an attempt to mitigate some of these three aspects and to avoid misleading results, data should be preventively treated prior to perform downstream analysis. In this chapter, two important count pre-processing steps are introduced, jointly with the most used and recent tools that address them. The chosen approaches include the complete set of pre-processing instruments that will be used in the following part of the thesis to compose the pre-processing pipelines on which the benchmark for optimal pre-processing practices will be performed.

3.1 Normalization

Normalization is the process of eliminating artifactual biases between samples, making possible a direct comparison of species abundance between them or between groups of them. Raw data can, in fact, contain peculiar biases due to sample collection, library preparation and sequencing that can imply uneven sampling depth and sparsity. Many downstream analyses, such as ordination analysis and statistical testing performed to look for specific bacteria that are differentially abundant between two ecosystems, may suffer from these experimental bias in a so heavy way that incorrect conclusions may be drawn if no correction is previously applied. A plethora of tools became available in last years for performing sequencing count data normalization. In this thesis, a subset of them has been selected for performance testing, these tools being the most vastly used or the most recent and promising now available. This collection is formed by approaches that nowadays we can call "historical", such as Total Sum Scaling, by more recent techniques, such as Cumulative Sum Scaling and the ones implemented in *edgrR* and *DESeq2 R* packages, and by two more recently developed tools, *scran* and *GMPR*, that directly address data heavily affected by sparsity.

3.1.1 Methods and tools for normalization

In this section, the six normalization approaches and tools chosen for normalizing count data will be reported. The subsection titles chosen to introduce each method/tool reflect the most used names to identify each approach and will be maintained as their labels all along this thesis. The name may be both the method name or the name of the *R* package in which it was first implemented, depending on which is the most familiar and used within users' community and specific literature.

3.1.1.1 Total sum scaling (TSS)

Total sum scaling (TSS) or global scaling [69] is the simplest and oldest way of normalizing sequencing data. It simply divides raw counts by the total number of reads found in the sample (the sequencing depth), i.e. it transforms count vectors in the corresponding vectors of proportions within the sample by simply computing $p_{ij} = c_{ij}/N_j$, where c_{ij} are the counts of *i*th feature in *j*th sample and N_j is *j*th sample sequencing depth. Because of its simplicity, this method was not taken from any implemented *R* package, but was directly calculated on raw data with basic *R* functions.

3.1.1.2 Cumulative sum scaling (CSS)

Cumulative sum scaling (CSS) performs a quantile normalization by looking for a data-specific quantile to use in order to normalize data in a coherent way. This method has been introduced by Paulson et al. [70] and then included in *metagenomeSeq* [71] *R* package, but literature always refer to it as CSS and so the same will be done in this thesis. The developers propose this normalization technique to correct for sequencing bias, that is thought to come from features that are preferentially amplified in a sample-specific manner. If we denote by q_j^l the lth quantile of sample j and by s_j^l the sum of counts for sample j up to the lth quantile, that is

$$s_j^l = \sum_{i|c_{ij} \leq q_j^l} c_{ij}, \tag{3.1}$$

the normalization method chooses a value $\hat{l} \leq m$, where m is the total number of features, to define a normalization scaling factor for each sample to produce normalized counts:

$$\widetilde{c_{ij}} = N \frac{c_{ij}}{s_j^{\hat{l}}}, \tag{3.2}$$

where N is an appropriately chosen constant applied to all samples so that normalized counts have interpretable units. The authors suggest this N to be chosen as the median of scaling factors $s_j^{\hat{l}}$ across samples. The choice of the appropriate value for $\hat{l}$ is crucial for ensuring the normalization approach does not introduce artifacts in the data. To determine the most appropriate value for $\hat{l}$, an adaptive, data-driven method is used that finds a value $\hat{l}$ for which sample-specific count distributions deviate from an appropriately defined reference distribution. In particular, if we consider $\bar{q}^l = med_j\{q_j^l\}$ the median lth quantile across samples as the lth quantile of the reference distribution and denote $d_l = med_j|q_j^l - \bar{q}^l|$ the median absolute deviation of sample-specific quantiles around the reference, $\hat{l}$ can be identified as the smallest value for which high instability is detected in high quantiles of d_l. Specifically, $\hat{l}$ is set to the smallest l that satisfies $d_{l+1} - d_l \geq 0.1 d_l$. The value 0.1 is set arbitrarily from the authors, but may be substituted by another value to determine high instability.

3.1.1.3 edgeR

edgeR's normalization procedure, namely "trimmed mean of M-values normalization method", is one of the most famous and widely used normalization techniques in sequencing data pre-processing. Its native framework was bulk RNA-Sequencing, but it has been used in a

variety of other situations involving sequencing count data, such as metagenomic and Single-Cell RNA-Sequencing studies. Robinson and his collaborators proposed [72] an empirical strategy that equates the overall expression levels of features between samples under the assumption that the majority of them are not differentially abundant. For sequencing data, they define the feature-wise log-fold-changes as:

$$M_g = log_2 \frac{Y_{ik}/N_k}{Y_{ik'}/N_{k'}} \tag{3.3}$$

and absolute presence levels as:

$$A_i = \frac{1}{2} log_2(Y_{ik}/N_k \cdot Y_{ik'}/N'_k), \qquad for \quad Y_{i\cdot} \neq 0 \tag{3.4}$$

Normalization factors are calculated by selecting one sample as a reference and calculating the trimmed mean of M-values as factor for each non-reference sample, as follows. A trimmed mean is the average after removing the upper and lower x% of the data. The trimmed mean of M-values (TMM) used in *edgeR*'s procedure is doubly trimmed, by log-fold-changes M^r_{ik} (sample k relative to sample r for feature i) and by absolute intensity (A_i). By default, the tool trims the M_i values by 30% and the A_i values by 5%, but these settings can be tailored to each experiment. After trimming, a weighted mean of M_i is computedand the normalization factor for sample k using reference sample r is calculated as:

$$log_2(TMM^r_k) = \frac{\sum_{i \in I^*} w^r_{ik} M^r_{ik}}{\sum_{i \in I^*} w^r_{ik}}, \tag{3.5}$$

where

$$M^r_{ik} = log_2 \frac{Y_{ik}/N_k}{Y_{ir}/N_r} \tag{3.6}$$

and

$$w^r_{ik} = \frac{N_k - Y_{ik}/N_k}{N_k Y_{ik}} + \frac{N_r - Y_{ir}/N_r}{N_r Y_{ir}}, \qquad Y_{ik}, Y_{ir} \geq 0. \tag{3.7}$$

The cases where $Y_{ik} = 0 or Y_{ir} = 0$ are prior trimmed since log-fold-changes cannot be calculated; I^* represents the set of features with valid M_i and A_i values and not trimmed, using the percentages above.

3.1.1.4 DESeq2

Another very well-known tool for RNA-Sequencing analysis is DESeq2 [73], which performs count data normalization and differential analysis. Also in this case, normalization is done through data scaling for sample-specific size factors. To estimate these size factors, *DESeq2*

package offers the median-of-ratios method already used in its first version, *DESeq* [74]. Following this method, size factors s_j for each sample are estimated as:

$$\hat{s}_j = median_i \frac{k_{ij}}{\left(\prod_{v=1}^{m} k_{iv}\right)}, \tag{3.8}$$

where k_{ij} are the observed counts for feature i in sample j. The denominator of this expression can be interpreted as a pseudo-reference sample obtained by taking the geometric mean across samples. Thus, each size factor estimate $\hat{s}_j$ is computed as the median of the ratios of the jth sample's counts to those of the pseudo-reference.

3.1.1.5 scran

This normalization method, recently introduced by Lun et al. [75], has its roots in the field of Single-Cell RNA-Sequencing (scRNA-Seq) studies and its birth was primarily due to the observation that classic RNA-Sequencing methods, such as *edgeR* and *DESeq*, would not be the most appropriate normalization tools for scRNA-Seq data, where the high frequency of dropout events and, consequently, the high sparsity of count data interferes with stable normalization. A possible approach could be removing the features most affected by zero values during normalization, but this may introduce biases if the number of zeroes varies across samples. Although this method was not directly thought for 16S count data pre-processing, we decided to include it because it addresses the same main problem that afflicts 16S data, i.e. heavy sparsity, and because its methodology does not rely on specific scRNA-Seq data properties that could make conclusions invalid for 16S data. In fact, *scran* normalization simply pools multiple samples in order to estimate sample-specific size factors more robustly in the presence of huge sparsity by introducing a deconvolution strategy: normalization factors calculated on summed expression values from pools of samples are secondly deconvolved into the size factors for its constituent samples. In particular, the deconvolution method consists of several key steps:

- Defining a pool of samples and summing expression values across all samples in the pool. This is done by ordering samples by their total counts and partitioning them into two groups, depending on whether the ranking of each sample is odd or even. These samples are arranged in a ring, with odd samples on the left and even samples on the right. Conceptually, one can start at the 12 o'clock position on the ring, for the largest libraries, move clockwise through the even samples with decreasing library size, reach the smallest libraries at 6 o'clock, and then continue to move clockwise through the odd samples with increasing library size. For summation, a sliding window is moved

sample-by-sample across this ring where each window contains the same number of samples. These samples are used to define a single pool.

- Normalizing the pool against an average reference, obtained using the mean of expression values. The normalization factor is obtained as the sum of proportional abundances for each sample in the pool divided by the mean of counts across all the samples in the entire dataset.
- Repeating this for many different pools of samples (through the sliding window) to construct a linear system in which every a pool-based size factor is equal to the sum of the sample-based factors of the pool constituent samples
- Deconvolving the pool-based size factors to their sample-based counterparts by standard least-squares methods

To facilitate the comprehension of the entire mechanism, we report here the summarizing graphical scheme included in the paper (Figure 3.1).

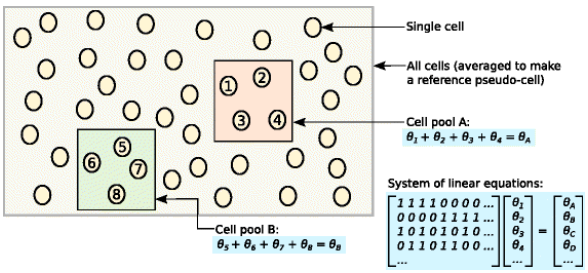


Fig. 3.1 Schematic representation of the deconvolution method. All samples in the data set are averaged to make a reference. Abundance values for samples in pool A are summed together and normalized against the reference to yield a pool-based size factor θ_A . This is equal to the sum of the sample-based factors θ_j for samples $j = 1,\ldots,4$ and can be used to formulate a linear equation. Repeating this for multiple pools (e.g., pool B) leads to the construction of a linear system that can be solved to estimate θ_j for each sample j. Image taken from [75].

3.1.1.6 GMPR

GMPR is a very recently published tool [76] that proposes a novel inter-sample normalization method, geometric mean of pairwise ratios (GMPR), developed specifically for zero-inflated sequencing data such as microbiome sequencing data (16S). This method extends the idea of *edgeR* normalization for RNA-Seq data and relies on the same assumption that there is a large invariant part in the count data. Assuming to have a count table of OTUs from 16S rDNA targeted microbiome sequencing, we denote as c_{ki} the count of the kth OTU ($k = 1,\ldots,q$) in the ith ($i = 1,\ldots,n$) sample. In *edgeR* method, since geometric mean is not well defined for

features with zeros, features with zeros are usually excluded in size calculation. However, for zero-inflated data such as microbiome sequencing data, as the sample size increases, the probability of existence of features without any zeros becomes smaller. In such cases, *edgeR* may be not adequate to pursue this purpose. As an alternative, adding a pseudo-count such as 1 or 0.5 to count tables to eliminate null values is a broadly used strategy. Since the majority of the counts may be zeros for microbiome data, adding even a small pseudo-count could have a dramatic effect on the geometric means of most OTUs. To circumvent the problem, *GMPR* reverses the order of the two steps present in *edgeR* workflow. The constitutive steps of the normalization process are then:

1. calculate r_{ij}, i.e. the median count ratio of nonzero counts between sample i and j:

$$r_{ij} = \underset{k \in \{1,\dots,q\} | c_{ki} \cdot c_{kj} \neq 0}{median} \left\{ \frac{c_{ki}}{c_{kj}} \right\} \tag{3.9}$$

2. calculate the size factor s_i for a given sample i as

$$s_i = \left(\prod_{j=1}^{n} r_{ij} \right)) \frac{1}{n}, \qquad i = 1, \dots, n. \tag{3.10}$$

Based on this analysis strategy, the tool utilizes far more information than TMM, that is usually restricted to a small subset of OTUs.

3.2 Zero imputation

As introduced in Chapter 2, 16S rDNA-Sequencing data analysis is typically complicated by excess zero counts. These null values may rise from a multiplicity of factors, but they may be attributed to two main sources: a biological and a technical one. Biological zeros are those null values present into a sample count distribution that represent features (OTUs, species,...) that are effectively not present in the sample. These zeros are constitutive of each sample population profiling and represent a true information of absence of some species within the sample. On the contrary, technical zeros are those null values that characterize unseen features within the sample, that is features that were present in the sequenced population but which information got lost during sequencing procedure due to their low abundance respect to other sample components. This loss of information may occur in different steps of a sequencing study: during amplification, library preparation and normalization and, of course, during the proper sequencing step. Zero-imputation pre-processing step arose precisely to

address this issue, i.e. for imputing missing values and restoring the structure of the data present before sequencing. The application of this step in sequencing frameworks where the zero problem was present, such the above-mentioned scRNA-Seq, demonstrated that although a part of the technical bias is not recoverable *a posteriori*, such that the portion due to amplification or library preparation, it is possible to efficiently recover information that got lost in the sequencing step by borrowing information from samples or features in which the information was luckily preserved. Even if in 16S rDNA-Seq this step has not been introduced yet, we believe that its insertion in the pre-processing workflow would bring appreciable improvements in microbiome analyses, especially in those studies in which rare species, i.e. the features that most probably would get lost in the sequencing step, are a central topic of research.

In this work, a collection of zero-imputation tools was considered for insertion in the pre-processing procedure before data downstream analysis performance. These tools were all developed for unseen information recovery, but in vary different frameworks, such as scRNA-Seq, microarray data analysis or even within the general matrix completion competition known as the Netflix problem.

3.2.1 Methods and tools for zero-imputation

In this section, six different zero-imputation approaches will be presented. Five of these have been used for pre-processing pipeline formation, while the last one was finally excluded for technical reasons that will be explained in the following. As for normalization methods, the subsection titles reflect the names with which each approach is publicly most represented and will be used as labels all along this thesis. The name may then represent both the method or the the *R* package name in which it was implemented, depending on which is the most familiar and used within user's community and specific literature.

3.2.1.1 DrImpute

DrImpute [77] is a just published zero-imputation tool for scRNA-Seq data that recovers information about null values imputing them by borrowing information from similar samples starting from normalized and log-transformed count data. *DrImpute* first identifies similar samples based on clustering, and imputation is then performed by averaging the values from similar samples. To achieve robust estimations, the imputation is performed multiple times using different sample clustering results followed by averaging multiple estimations for definitive imputation. First, the sample-sample distance matrix is computed using Spearman and Pearson correlations, followed by the sample-wise clustering based upon the distance

matrix over a range of expected number of clusters k (k ranging from 10 to 15 by default). For each combination of distance metric (Spearman or Pearson) and k, the recovered values in the input matrix are estimated. The averaged estimation over all combinations form then the final imputed values. In particular, let X be a n by p log-transformed count matrix, where n is the number of rows (features) and p is the number of columns (samples) and H be the number of clustering configurations (e.g. combinations of distance metric and number of clusters). If $C_1, C_2, \ldots, C_h$ are the related clustering results, the expected value of a false zero (dropout) event can be obtained by averaging the entries in the given cluster:

$$E(x_{ij}) = mean(x_{ij} | x_{ij} \text{ in the same group in clustering } C_h). \tag{3.11}$$

This value is computed for each clustering result $C_1, C_2, \ldots, C_h$ and the final imputation for x_{ij} is computed as a simple averaging:

$$E(x_{ij}) = mean(E(x_{ij}|C)) = \frac{1}{H} \sum_{h=1}^{H} E(x_{ij}|C_h). \tag{3.12}$$

The clustering step is made by a K-means clustering on the first 5% of the principal components of the similarity (Spearman, Pearson) matrix.

A scheme of this step as included in the Supplementary material of the original article [77] is reported in Figure 3.2. We recall that, for generalization purpose, "gene" could be substituted with "feature" and "cell" with "sample".

3.2.1.2 scImpute

Also this tool was published very recently [78], in 2018, and for scRNA-Seq pre-processing. One interesting feature of this method is that it automatically identifies likely dropouts, and only performs imputation on these values without introducing new biases to the remaining data. To achieve this goal, *scImpute* first learns each feature's dropout probability in each sample based on a mixture model. Next, it imputes the (highly probable) dropout values in a sample by borrowing information of the same feature in other similar samples, which are selected based on the features unlikely affected by dropout events and then labelled as reliable source of information. To help the reader, we report here (Figure 3.3) a toy example present in the paper [78] before going into details with the mathematical description of the workflow. Also in this case, to tranfer the scheme from scRNA-Seq framework to a general one, we could think of "cells" as "samples" and "genes" as "features".

The input matrix asked by *scImpute* tool is a raw count matrix X. Normalization step is then internally performed with simple TSS normalization (division by sample library size).

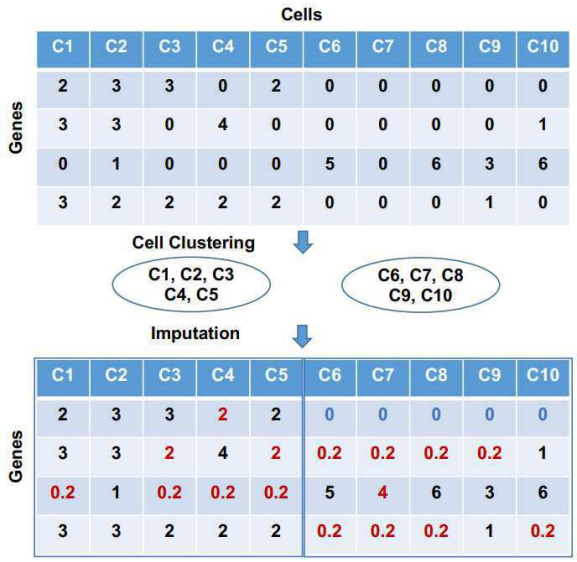


Fig. 3.2 Basic procedure for clustering-based imputation. Upper matrix is a gene by cell matrix. After clustering on gene by cell matrix, we observe $C1-C5$ as one cluster and $C6-C10$ as the other cluster. Imputation is performed by averaging each cluster. Figure taken from [77].

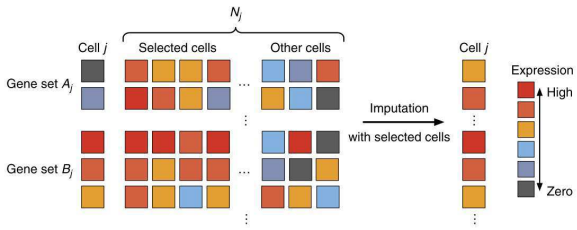


Fig. 3.3 A toy example illustrating the workflow in the imputation step of *scImpute* method. *scImpute* first learns each gene's dropout probability in each cell by fitting a mixture model. Next, it imputes the (highly probable) dropout values in cell j (gene set A_j) by borrowing information of the same gene in other similar cells, which are selected based on gene set B_j (not severely affected by dropout events). Image taken from [78].

The normalized matrix (X^N) is then added with a pseudo-count 1.01 and log-transformed:

$$X_{ij} = log_{10}(X_{ij}^N + 1.01), \quad i = 1,\ldots,I; j = 1,\ldots,J, \tag{3.13}$$

Where I is the total number of features and J the total number of samples. The imputation is then performed into three steps:

- **Detection of sample subpopulations**. Since scImpute borrows information of the same feature from similar samples to impute the null values, a critical step is to first determine which samples are from the same subpopulation. This is done by performing a Principal Component Analysis (PCA) on the matrix X and selecting PCs such that at

least 40% of the variance in data could be explained. On this PCs, a distance matrix is computed and samples are then clustered into K groups by spectral clustering.

- **Identification of dropout values**. Each feature is modelled with a mixture model of two components: a Gamma distribution used to account for zeros and a Normal distribution to represent the actual abundances. For each feature, its abundance level is a random variable with density function:

$$f_{X_i}^{(k)}(x) = \lambda_i^{(k)} Gamma\big(x; \alpha_i^{(k)}, \beta_i^{(k)}\big) + \big(1 - \lambda_i^{(k)}\big) Normal\big(x; \mu_i^{(k)}, \sigma_i^{(k)}\big) \tag{3.14}$$

where $\lambda_i^{(k)}$ is feature i's dropout rate in cluster k, $\alpha_{X_i}^{(k)}, \beta_{X_i}^{(k)}$ are the shape and rate parameters of Gamma distribution and $\mu_{X_i}^{(k)}, \sigma_{X_i}^{(k)}$ are the mean and standard deviation of Normal distribution. The intuition behind this mixture model is that if a feature has high abundance and low variation in most samples within a sample subpopulation (cluster), a zero count is more likely to be an unseen value; on the other hand, if a feature has constantly low or medium abundance with high variation, then a zero count may reflect real biological variability. The parameters in the mixture model are estimated by the Expectation-Maximization (EM) algorithm, thus obtaining the dropout probability of feature i in sample j, which belongs to subpopulation k:

$$d_{ij} = \frac{\hat{\lambda}_i^{(k)} Gamma\big(X_{ij}; \hat{\alpha}_i^{(k)}, \hat{\beta}_i^{(k)}\big)}{\hat{\lambda}_i^{(k)} Gamma\big(X_{ij}; \hat{\alpha}_i^{(k)}, \hat{\beta}_i^{(k)}\big) + \big(1 - \hat{\lambda}_i^{(k)}\big) Normal\big(X_{ij}; \hat{\mu}_i^{(k)}, \hat{\sigma}_i^{(k)}\big)}. \tag{3.15}$$

- **Imputation of null values**. For each sample j, fixed a threshold t on dropout probabilities, we divide features into two groups: $A_j = \{i | d_{ij} \geq t\}$ and $B_j = \{i | d_{ij} < t\}$. The first set is made of features in need of imputation, while the second one contains features with high confidence from which the model will learn information for imputation. To learn the samples similar to sample j from B_j, the non-negative least squares (NNLS) regression is used:

$$\hat{\beta}^{(j)} = \text{argmin}_{\,\beta^{(j)}} ||X_{b_j,j} - X_{b_j,N_j}\beta^{(j)}||_2^2, \quad \text{with } \beta^{(j)} \geq 0, \tag{3.16}$$

where:

  - N_j represents the indices of samples that are candidate neighbours (samples in the same cluster) of sample j
  - $X_{b_j,j}$ is a vector representing the B_j rows in the jth column of X
  - X_{b_j,N_j} is a sub-matrix of X with dimensions $|B_j| \times |N_j|$

– $\beta^{(j)}$ is a vector of length $|N_j|$.

Finally, the estimated coefficients $\hat{\beta}^{(j)}$ from the set B_j are used to impute the abundances of features in the set A_j in sample j:

$$\hat{X}_{ij} = \begin{cases} X_{ij}, & \text{i} \in B_j \\ X_{i,N_j}\hat{\beta}^{(j)}, & \text{i} \in A_j. \end{cases} \tag{3.17}$$

3.2.1.3 LLSImpute

Contrarily to *DrImpute* and *scImpute*, *LLSimpute* [79] tries to estimate missing values using information stored in co-abundant or similar features. It is designed based on a linear regression model, which divides samples into two groups (C_i and D_i) for feature i. C_i stores samples in need of imputation, i.e. with null values, while D_i collects the non-null values. Suppose there are q missing values for feature i, it finds the K-nearest neighbour feature vectors for feature i based on values in D_i, which may be represented as $G_{K_i,D_i} \in \mathbb{R}^{K\times(n-q)}$ where n is the total number of samples. Let $G_{i,D_i} \in \mathbb{R}^{1\times(n-q)}$ denote feature abundance of feature i across samples in D_i, then G_{i,D_i} may be represented as a linear combination of rows of G_{K_i,D_i} by:

$$min_x||G^T_{K_i,D_i}x - G_{i,D_i}||_2. \tag{3.18}$$

Then the missing values of feature i denoted by G_{i,C_i} are imputed by $G^T_{K_i,C_i}x$, where G_{K_i,C_i} represents abundance levels of features K_i across samples C_i.

3.2.1.4 LowRank

Low-rank method included into this thesis [80] is borrowed by the general matrix completion framework in which, for example, the Netlix problem is included. In that case, as users only rate a few items, one would like to infer their preference for unrated ones. Obviously, only a few factors affect an individual's preference and then the user-rating data matrix should be in low-rank. This method has been recently applied [81] to recover information in scRNA-Seq data, that have similar sparsity characteristics as 16S-seq data. Thus, we decided to test for its performance also in this context. Low-rank method supposes that the count matrix X without zeros is low-rank and can be approximated by its nuclear-norm, which is its convex envelope. The model then solves the problem of finding:

$$min_X||X||_* \tag{3.19}$$

such that

$$||x_\Omega - G||_F^2 \leq \delta, \tag{3.20}$$

where X is the imputed count matrix, G is the raw one, Ω is the observed space, δ is the tolerance between the imputed data and the observed one and $||\cdot||_F$ is the Frobenius norm.

3.2.1.5 zCompositions

zCompositions [82] is linked to (and is a product of) a research on Bayesian tools for count zeros in compositional data sets [83] [84], i.e. those datasets composed by discrete vectors representing the numbers of outcomes falling into any of several mutually exclusive categories. As a sampling process, 16S rDNA sequencing produces count data that perfectly fall into this definition. For this reason, this (and the following) tool has been considered for pre-processing treatment in this thesis. Martin-Fernandez and collaborators propose [85] a Bayesian imputation method for zero counts based on multiplicative replacement, starting from a typical multinomial modelling of count data and a Dirichlet distribution as its conjugate prior. Using this methodology one can retrieve lost values preserving the ratios between non-zero components in the samples. This strategy consists on replacing the zero and non-zero proportions for each sample as follows:

$$\begin{cases} x_j^* = \frac{\alpha_j}{s+N}, & x_j = 0, \\ x_j^* = x_j(1 - \sum_{x_k=0} \frac{\alpha_k}{s+N}), & x_j > 0, \end{cases} \tag{3.21}$$

with N the feature number and $\alpha = (\alpha_1, \ldots, \alpha_D) = s \cdot t = s \cdot (t_1, \ldots, t_D)$, where s is the strength of the prior and $t = (t_1 \ldots, t_D)$ refers to the prior estimates of the multinomial probabilities. In this model, all zero percentages are replaced by its posterior expectation and the non-zero percentages are multiplied by a factor according to the number of zero counts. In the tool, several multiplicative Bayesian imputations are implementedthat differ both for the prior distribution used to model the random vector of multinomial probabilities and for the replacement of zero values. Among these, in this thesis three different approaches that the authors consider for comparison were selected, as in the following.

- Geometric Bayesian Multiplicative (GBM) approach. Suppose we want to replace zeros in vector x_i. Let then α_{ij} be the sum of counts $c_{\cdot j}$ for feature j excluding the ith sample, i.e.

$$\alpha_{ij} = \sum_{K=1; k \neq i} c_{kj}. \tag{3.22}$$

Let now introduce $\hat{m}_{ij}$ defined as:

$$\hat{m}_{ij} = \frac{\alpha_{ij}}{\sum_{k=1}^{D} \alpha_{ik}}. \tag{3.23}$$

GBM method is obtained by using $\hat{m}_{ij}$ as prior estimate for multinomial probabilities.

- Square root (SQ) Bayesian Replacement. This method uses as prior parameters $s = \sqrt{n}$, where n is the multinomial number of trials, and a uniform prior distribution (t), that is $t = (\frac{1}{D}, \ldots, \frac{1}{D})$.
- Rounded zero multiplicative replacement (CZM). According to this approach, each zero in the proportion vector is replaced by the small value $0.65 \cdot 0.5/n$ and, then, the non-zero values are modified in a multiplicative way.

Despite GMB good performance highlighted in the paper [83], this *zCompositions* configuration had to be excluded from the benchmark. In fact, the implementation of this approach resulted to be affected by the fact that some features may appear only in a unique sample throughout the entire count data matrix, a situation that is not infrequent in 16S rDNA-Sequencing data analysis. All datasets used for benchmark procedure had indeed this typical characteristic and the only alternative for using this tool was to delete some feature *ad hoc*. Being the only tool not able to manage this situation, it was considered a more fair approach to perform the benchmarking excluding this tool and not deleting information for all the study to make it work. Regarding SQ and CZM approaches, they were included in the benchmark and we will refer to them throughout all the thesis as *zCompositions_SQ* and *zCompositions_CZM*.

3.2.1.6 robCompositions

Another compositional approach considered in this thesis is implemented in *robCompositions* *R* package [86]. This set of functions implements the methods introduced by Hron, Temple and Filzmoser in their work [87], where they propose two different imputation algorithms for estimating missing values in compositional data: a k-nearest neighbour (knn) imputation and an iterative model-based imputation. In the following the details about the two techniques are reported.

knn imputation.The knn imputation uses a distance measure for finding the k most similar samples to a given one containing zeros, and to replace the missings by using the available variable information of the neighbours. In the context of compositional data,

Aitchison distance [88] is commonly used to calculate sample-sample distances. In formulae, given two generic samples x and y, their Aitchison distance is calculated as:

$$d_A(x,y) = \sqrt{\frac{1}{D}\sum_{i=1}^{D-1}\sum_{j=i+1}^{D}\left(ln\frac{x_i}{x_j} - ln\frac{y_i}{y_j}\right)}. \tag{3.24}$$

For imputing a missing value of a sample the median of the corresponding samples of the k-nearest neighbours is used. More formally, if we denote by $M_i \subset \{1,\ldots,D\}$ the set of indices referring to the null values within a sample, and with O_i the remaining ones, for imputing a missing value x_{ij} we consider among all other samples those which have non-missing values at position j and O_i and compute the k-nearest neighbours of sample x_i using Aitchison distance. To make the k-nearest neighbors comparable, observations have to be scaled by factors comparing the size of the parts in O_i and calculated as follows:

$$f_{ii_l} = \frac{median_{o\in O_i} x_{io}}{median_{o\in O_i} x_{i_l o}}, \text{for } l \in \{1,\ldots,k\}. \tag{3.25}$$

Using these factors as weights, the imputed value replacing the missing one can be calculated as:

$$x^*_{ij} = median\{f_{ii_1}x_{i_1 j},\ldots,f_{ii_k}x_{i_k j}\}. \tag{3.26}$$

Iterative model-based imputation. In the iterative imputation method, at each step of the iteration one variable is used as a response variable and the remaining variables serve as the regressors. Thus the multivariate information is be used for imputation in the response variable. In detail, the iterative algorithm based on regression can be summarized in the following steps:

1. Initialize the missing values using the knn algorithm
2. Sort the variables according to the amount of missing values. Without loss of generality, we can think variables are already sorted following their index
3. Set $l = 1$
4. Use the ilr transformation seen in Section 2.2.1 to transform the compositional data set
5. denote $m_l \subset \{1,\ldots,n\}$ the indices of the observations that were originally missing (before knn inizialization) in sample x_i and o_l the remaining ones. Then, $z_l^{o_l}$ and $z_l^{m_l}$ denote the lth ilr transformations with the observed and missing features,

respectively, corresponding to the variable x_l. Let $Z_{-l}^{o_l}$ and $Z_{-l}^{m_l}$ be the matrices with the remaining z vectors corresponding to the observed and missing features of x_l, respectively, with the first columns composed by ones, taking care of an intercept term in the regression problem

$$z_l^{o_l} = Z_{-l}^{o_l}\beta + \varepsilon, \tag{3.27}$$

with unknown regression coefficients β and an error term ε

6. Estimate the regression coefficients β and use the estimation $\hat{\beta}$ to replace the missing features in $z_l^{m_l}$ by
$$\hat{z}_l^{m_l} = Z_{-l}^{m_l}\hat{\beta} \tag{3.28}$$
7. Perform a back-transformation to the simplex, thus updating originally missing values in m_l positions in sample x_l
8. Carry out Steps 4–7 in turn for each feature in $\{1,\ldots,D\}$
9. Repeat Steps 3–8 until the Euclidean distance between the empirical covariance matrices computed from the ilr-transformed data from the present and the previous iteration is smaller than a certain boundary.

Despite being very interesting and well-structured imputation strategies, the solutions proposed in *robCompositions* had to be excluded from this thesis benchmark. In fact, the knn imputation implementation showed not to manage situations in which too many features with null values are present in the anayzed dataset, a situation that is the most frequent one in 16S rDNA-Sequencing data analysis. This caused the impossibility of using knn imputed values to initialize the iterative algorithm, that was consequently also excluded. For this last, also a solution with "roundedZero" option was tried. This option (not exposed in the *R* function help, but indeed implemented) performs a zero-substitution with 0.001 to initialize the iterative procedure. This led no solution, because the high percentage of null values still represented a problem for regression methods, that all stopped or had to be stopped before hours of running time with no solution reached. This tool was consequently excluded from the comparison.

Chapter 4

Real data for simulator testing

Integral part of this thesis was the design of two metataxonomic studies in the field of Public Health and Food Safety. Data obtained from these two experiments were used for the simulator performance assessment and will be in the next future the first application context of the best performing pre-processing pipelines identified in this work. In this chapter, these two 16S experiments are described, with all the details from sample acquisition to final count matrix formation. In addition, the details about HMP data ([89, 90]), also used along this work, are reported. These datasets vary both in the sequencing platform adopted for their production (454 sequencing for HMP and Illumina HiSeq2500 for gut and food microbiome datasets) and in the characteristics of the final count matrix. In fact, the three datasets differ for sparsity levels, sequencing depth, number of samples and replicates. These datasets were used for metaSPARSim performance assessment and to perform the benchmark of 16S rDNA-Sequencing pre-processing pipelines. Synthetic datasets were created mimicking these real datasets.

4.1 Animal gut microbiome

This sample collection had the aim of monitoring chicken gut microbiota modifications in the first 4 weeks of life in occurrence of *Campylobacter* spp. infection, and comparing them with healthy chickens risen in analogous conditions but within farms in which no *Campylobacter* spp. infection occurred.

Experimental design. For this study, we selected four broiler farms belonging to the same supply chain, half of which became positive for *Campylobacter* spp. during the sampling period, while the other half showed no infection during all the 4 weeks of monitoring. Five samples were collected from each farm at 5 different time points (7th,

14th, 18th, 21th and 28th day of chickens' age), for a total of 110 caecal samples. For the two positive farms, ten samples (five positive and five negative for *Campylobacter* spp.) were collected for the time points in which infection was first detected. A scheme of this experiment is reported in Figure 4.1.

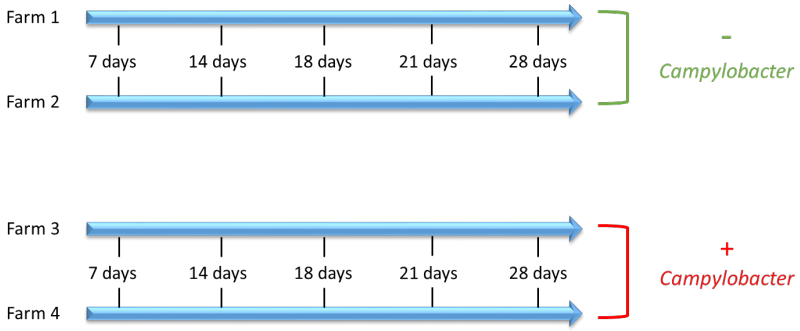


Fig. 4.1 Scheme representing the experimental design of animal gut microbiome study.

Sampling. Biological samples of caecal material were collected using the fecal swabs produced by Copan. The suitability of buffers to the study objective was verified through preliminary tests. Each buffer contains 2 ml of a transport medium called the modified Cary Blair.

DNA extraction. Total DNA was extracted using a column-based kit (QIAamp DNA Mini Kit, QIAGEN) starting from 200 µl of caecal content. Thermal lysis was carried out for 2 hours and RNase (100 mg/ml) was added to each sample to ensure RNA-free preparation. Total DNA was resuspended in 200 µl of nuclease-free water and stored at -20°C until preparation for sequencing. Two biological replicates were extracted for each sample.

16S rDNA sequencing. V3-V4 regions of 16S rRNA gene were amplified with the primers CCTACGGGNGGCWGCAG (forward) and GACTACHVGGGTATCTAATCC (reverse), following Klindworth et al.[91], and sequenced on HiSeq2500 platform in RAPID mode (2x250 bp).

Read pre-processing and count data formation. Sequencing data underwent a quality control procedure using the FastQC tool [92]. Data were then cleaned by removing adapters, primers and performing dereplication of sequences using a in-house bash script. In addition, data were filtered based on the quality and length of the reads, so that only data with a quality higher than a given threshold (QPhred ≥ 20) and reads whose length exceeded 100bp were retained. All subsequent steps were performed

using Python [93] scripts that are part of the QIIME1 [51] pipeline (version 1.9.0). Data obtained from the filtering step underwent read pairing, in order to obtain a single file in which the reads obtained by sequencing the 16S fragments on the forward strand and on the reverse are joined by their overlapping region. Then, OTU picking step was performed, assigning reads to a particular taxonomy by directly mapping the same reads to a 16S sequences database (GreenGenes database [56], last release May 2013).

Pre-estimation filtering Before parameter estimation and synthetic data simulation, a filtering procedure was performed to eliminate singletons, i.e. sequences that are observed only once. In fact, it has been shown ([94],[95],[96]) how most singletons in sequencing data result from DNA sequencing errors, such as base substitutions, base deletions, low-quality reads, variable read lengths and nontarget amplification. These errors, together with the presence of undetected chimeric sequences, caused by the hybridization of DNA fragments from different species, imply the formation of singletons and, thus, the creation of false OTUs.

4.2 Food microbiome

The second study had the aim of following the dynamics of the microbial community of "Latteria" raw milk cheese during its ripening period, in natural ageing conditions and in presence of contamination by pathogens like *Listeria monocytogenes* and *Staphilococcus aureus*.

Experimental design. The cheesemaking was made in a dairy in Friuli Venezia Giulia region, following all the typical steps of this particular raw milk cheese production. The design was made by 4 types of cheesemaking:

- plain, with no contamination
- contaminated with *Listeria innocua*, chosen for security reasons as a substitute of *Listeria monocytogenes* as being characterized by the same dynamical behaviour
- contaminated with *Staphilococcus aureus*
- contaminated with both *Listeria innocua* and *Staphilococcus aureus*.

Each cheesemaking was performed in triple replicate, to access biological variability. Samples from from cheese till the 30rd day of ripening period were collected, for a total of 10 sampling time points and 120 samples (12 from cheese at each time point). A scheme of this experiment is showed in Figure 4.2.

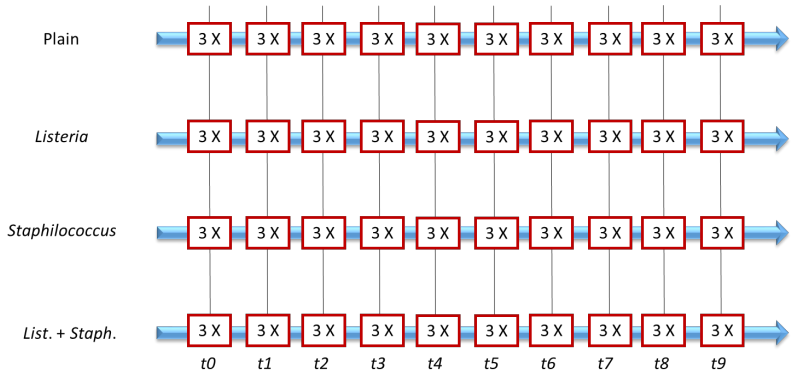


Fig. 4.2 Scheme representing the experimental design of animal gut microbiome study.

Sampling. Cheese aliquots were collected by a core sampler applied on the heel of the cheeses. At the end of the withdrawal, the holes inside the forms were sealed with a sterile putty, specially formulated for cheese storage.

RNA extraction. For the cheese sampling, about 15g of product was extracted from each form and placed in RNAlater® (Sigma-Aldrich), a preservation solution that allows to stabilize and protect cellular RNA. 50mg of initial matrix were taken from each sample and placed in 2ml Eppendorf tubes for total RNA extraction. RNA extraction was performed with Power Lyzer UltraClean Tissue & Cells RNA Isolation Kit (MOBIO Laboratories). After homogeneization and centrifugation (13000 r/min for 1 minute), the supernatant was collected and the sample was washed to obtain clear RNA. Then, DNase was added to each sample to ensure DNA-free preparation. Then, RNA reverse transcription was performed using SuperScript IIReverseTranscriptase (Invitrogen) kit.

16S rcDNA sequencing. V3-V4 regions of 16S rRNA gene were amplified with the primers CCTACGGGNGGCWGCAG (forward) and GACTACHVGGGTATCTAATCC (reverse), following Klindworth et al.[91], and sequenced on HiSeq2500 platform in RAPID mode (2x250 bp).

Read pre-processing and count data formation. After sequencing, a control procedure using the FastQC tool [92] was performed on resultant data. After that, QIIME2 [51] pipeline (version 2017.11) was used to perform sequence quality control and feature table construction via DADA2 pipeline, which also performs phiX reads and chimeric sequences filtering. Taxonomic assignment was obtained using the QIIME2 Naive Bayes classifier pre-trained on Silva [58] database.

Pre-estimation filtering Analogously to what was done for the gut microbiome dataset, a filtering procedure was performed to eliminate singletons before parameter estimation

and data simulation. Two samples were excluded from the dataset due to the low sequencing depth.

4.3 Human Microbiome Project data

As above introduced, also HMP data ([89, 90]) were used to test for metaSPARSim performance. The project had the aim of creating resources to easily characterize the human microbiota. Within this project, the microbial communities from 300 healthy individuals across several different sites on the human body was characterized: nasal passages, oral cavity, skin, gastrointestinal tract, and urogenital tract samples were collected and variable regions 3-5 (V35) of 16S rRNA gene were sequences (Figure 4.3). The datasets considered in this thesis thatr were derived from HMP dataare composed by a subsample of this publicly available dataset, taking 5 or 100 random samples from 8 sampled groups from oral cavity: hard palate, gingivae, tongue dorsum, teeth, palatine tonsils, throat, saliva and buccal mucosa.

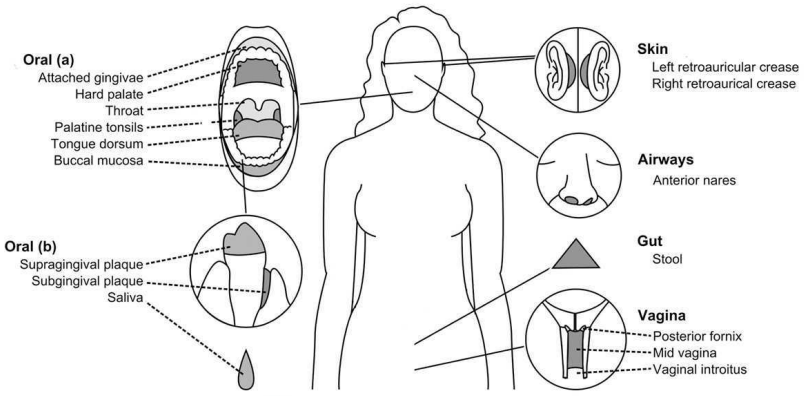


Fig. 4.3 Human Microbiome Project sampling map. Adapted from [97]

The presented data were chosen to represent a good variety of data types a scientist may have to handle dealing with 16S rDNA studies. In addition to basic characteristics differences (Table 4.1) data are, for experimental construction, very different one from each other. In fact, what we called "replicates" in animal gut or food data were real biological replicates, i. e. different samples of the same material (or a clone of) at the same time point. For animal gut microbiota, "replicates" were samples taken at the same place and time from clonal chickens, i.e. chicks that had been bred in the same shed, eating the same food and living in the same place only with chickens equal in age. For cheeses, the replicates came from different but identically produced and ripened wheels. Conversely, data produced by the Human Microbiome Project are not obtained by repeatedly sampling the same individual,

Table 4.1 Real datasets characteristics.

Characteristic \ Dataset	**Animal gut**	**Raw milk cheese**	**HMP (small)**	**HMP (big)**
Samples	110	118	40	800
Replicates	5	2-3	5	100
Groups	22	40	8	8
Features	3541	3109	758	1767
Sequencing depth (range)	88692-832309	28536-349754	2798-24095	1010-49414
Sample material	16S rDNA	16S rRNA	16S rDNA	16S rDNA
Count data sparsity	78.69%	97.12%	81.37%	91.77%

but collecting several samples associated by what we can call "experimental condition". In particular, the 16S data from this study are derived from samples taken from different people in the same body sites. This causes the replicates included in this dataset to be characterized by a higher variability. This feature permitted to test the simulator performance in case of low (animal gut microbiota), medium (raw milk cheese experiment) and high (HMP) variance replicates, thus completely addressing one of the main aspects of data characterization in microbiome studies.

Chapter 5

metaSPARSim: a 16S count matrix simulator

As pointed out in the introduction, in the last few years 16S rRNA gene sequencing has seen a surprisingly rapid increase in election rate as a methodology to perform microbial community studies. Despite the considerable popularity of this technique, an exiguous number of specific tools are currently available for amplicon metagenomic data pre-processing that consider their sparse and compositional nature (see Chapter 2). Thus, developing optimal pipelines that fill this gap is of preeminent importance to assure solid and reliable conclusions from metagenomic data analyses. In order to achieve this goal, synthetic data simulators that allow for tool testing in a controlled situation are also needed. To the best of our knowledge, available 16S sequencing data simulators permit to obtain simulated reads from an hypothetical 16S experiment, but do not directly produce the count tables that are the main object developers of count data pre-processing tools use in their research.

For the above reasons, when working on identifying the optimal pre-processing pipeline for 16S count data it became pivotal to first develop a sparse matrix simulator that was able to output realistic count data matrices on which testing the performance of pre-processing tools and combinations of them. In this chapter, the result of this research topic is shown.

5.1 Count data modeling

As introduced in the previous chapter (Section 2.1.4, Table 2.1), after a read processing pipeline NGS sequencing data are finally summarized into a count table, a quantitative representation of presence of each feature within each sample. In the past years, several models were proposed to describe the nature of this type of data. In the following, the

most used modelling frameworks will be reported, to finally introduce the one specifically designed in the implementation of *metaSPARSim* simulator for 16S count data.

5.1.1 Poisson model

The classical approach for modelling a count random variable Y_{ij} is the univariate Poisson distribution, whose probability mass function (PMF) is expressed as:

$$P_{Pois}(y_{ij}|\lambda_{ik}) = \frac{{\lambda_{ik}}^{y_{ij}} e^{-\lambda_{ik}}}{y_{ij}!}, \qquad y_{ij} \in \{0,1,2,\dots\} \tag{5.1}$$

where λ_{ik} is the standard rate parameter and stands for the mean expected count value for feature i in experimental group k to which sample j belongs. This framework is based on the fact that a DNA sample can be seen as a collection of fragments taken from the species present within it and then DNA sequencing can be compared to a random sampling of the species, with the aim of estimating the relative abundance of each species in the niche. If we think each cDNA fragment like having the same chance of being selected for sequencing and the fragments being selected independently, then the number of counts for a given feature in repeated measurements could be described with a Poisson variation law. The above model has a well known main characteristic, that is the mean equals the variance. The consistence of this hypothesis has been examined in Marioni et al.[69], in which the same initial collection of RNA distributed across multiple lanes of Genome Analyzer (Illumina) sequencer was used. In this work, the Poisson model turned out to be a good description for technical replicates for most of the features, despite some other seemed to exhibit greater variability levels. However, no biological or extra, technical variability was included in that validation that consider the effect of sequencing different samples (i.e. samples with different internal count distribution along features) together. In fact, this causes the count mean to show higher variance than mean, i.e. the observation of the so-called over-dispersion phenomenon, that needs to be taken into account.

5.1.2 Negative Binomial distribution

To describe the biological plus technical variance, another well established model based on the Negative Binomial (NB) distribution has be adopted in sequencing count data analysis. The NB arises as a compound probability distribution where the distribution of the Poisson rate λ_{ik} is described by a gamma distribution, which is why NB also called Poisson-Gamma mixture distribution. Due to the extra-variation introduced by the gamma component, the resulting distribution then acts like an over-dispersed Poisson model. In particular, Robinson,

McCarthy and Smyth in their work [72] assume a negative binomial distribution for the read counts Y_{ij} for all genes, that is:

$$Y_{ij} \sim NB(\lambda_{ij}, \phi_i) \tag{5.2}$$

where

$$P_{NB}(y_{ij}|\lambda_{ik}, \phi_i) = \frac{\Gamma(y_{ij} + \frac{1}{\phi_i})}{\Gamma(y_{ij}+1)\Gamma(\frac{1}{\phi_i})} \left(\frac{1}{1+\lambda_{ik}\phi_i}\right)^{\frac{1}{\phi_i}} \left(\frac{\lambda_{ik}\phi_i}{1+\lambda_{ik}\phi_i}\right)^{y_{ij}} \tag{5.3}$$

and λ_{ik} is the mean of Y_{ij} in experimental group k to which sample j belongs while ϕ_i is the dispersion parameter for feature i. Under this definition, the total variance of Y_{ij} would be:

$$Var(Y_{ij}) = \lambda_{ik}(1 + \lambda_{ik}\phi_i). \tag{5.4}$$

Expressing Y_{ij} as mixture distribution, we have:

$$\begin{aligned} Y_{ij} &\sim Poisson(\lambda_{ik}), \\ \lambda_{ik} &\sim gamma(\frac{1}{\phi_i}, \lambda_{ik}\phi_i). \end{aligned} \tag{5.5}$$

When dispersion parameter ϕ_i goes to zero, the variance of Y_{ij} equals its mean. Recalling that the mean of a gamma distribution is the product of shape and scale parameters (here $\frac{1}{\phi_i}$ and $\lambda_{ik}\phi_i$, respectively) and its variance is the mean multiplied by the scale, the null value of ϕ also implies that $mean(\lambda_{ik}) = \lambda_{ik}$ and $Var(\lambda_{ik}) = 0$. That is, the rate parameter in the mixture distributions is not allowed to vary anymore, thus obtaining again the simple Poisson distribution.

5.1.3 Accounting for extra zero counts: zero-inflated and hurdle models

As described in Chapter 2, typical data in a 16S microbiome study consist of the operational taxonomic unit (OTU) counts, which have the characteristics of being non-negative, over-dispersed, compositional and having a much larger than expected number of observed zeros than assumed by Poisson or Negative Biomial distribution (up to ∼90-95% of total values). This phenomenon is known as "zero-inflation". As recalled in Xu et al. [60], one way to deal with such a big amount of zeros in count data is to use zero-inflated (ZI) models [98], which are basically mixtures of Poisson or Negative Binomial models with a point mass at zero.

Another approach is to use a hurdle model [99], a model formed by two parts, the first being modelled by a binomial distribution used to determine whether a zero or non-zero outcome occurs and the second being a count data modelling truncated at zero to characterize positive counts. The main difference between the two approaches lies in the fact that ZI models assume that the zero observations have two different origins: “structural” and “random”. The first zero values are real zeros that indicate the absence of the feature in the sample. On the other hand, random zeros are caused by a sampling problem, mainly due to insufficient depth when performing sequencing, thus causing rare taxa to be dropped from the sequenced population. On the contrary, hurdle models do not make the distinction between structural and sampling zeros, assuming that all zero data are from one unique “structural” source and thus treating them identically.

Zero inflated models

The most used zero-inflated models in sequencing count data are Zero-Inflated Poisson (ZIP) and Zero-Inflated Negative Binomial (ZINB) models. They both assume that for each observation there is a double possible generation process with the result of a Bernoulli trial switching between the one and the other. The first process generates only zero counts ("structural" zeros), while the second generates counts from a Poisson or NB distribution, respectively. If we denote as π_i the probability of structural zeros, then the probability function of Y_{ij} can be written compactly as:

$$f(Y_{ij}) = \pi_i I_0(Y_{ij}) + (1-\pi_i) f_{count}(Y_{ij}) \tag{5.6}$$

where $I_0(y)$ is a point mass at zero and $f_{count}(Y_{ij})$ follows Poisson or NB distribution.

More precisely, the **ZIP distribution** can be expressed as:

$$Y_{ij} \sim \begin{cases} 0 & \text{with probability } \pi_i \\ Poisson(\lambda_{ik}) & \text{with probability } (1-\pi_i) \end{cases} \tag{5.7}$$

or, following the previously introduced notations,

$$P_{Pois}(y_{ij}|\lambda_{ik},\pi_i) = \begin{cases} \pi_i + (1-\pi_i)e^{-\lambda_{ik}} & \text{if } y_{ij} = 0 \\ (1-\pi_i)\frac{\lambda_{ik}{}^{y_{ij}} e^{-\lambda_{ik}}}{y_{ij}!} & \text{if } y_{ij} > 0 \end{cases} \tag{5.8}$$

For the the **ZINB model**, we can write

$$Y_{ij} \sim \begin{cases} 0 & \text{with probability } \pi_i \\ NB(\lambda_{ij}, \phi_i) & \text{with probability } (1-\pi_i) \end{cases} \tag{5.9}$$

or, again using above notations,

$$P_{NB}(y_{ij}|\lambda_{ik}, \pi_i) = \begin{cases} \pi_i + (1-\pi_i)\left(\frac{1}{1+\lambda_{ik}\phi_i}\right)^{\frac{1}{\phi_i}} & \text{if } y_{ij} = 0 \\ (1-\pi_i)\frac{\Gamma(y_{ij}+\frac{1}{\phi_i})}{\Gamma(y_{ij}+1)\Gamma(\frac{1}{\phi_i})}\left(\frac{1}{1+\lambda_{ik}\phi_i}\right)^{\frac{1}{\phi_i}}\left(\frac{\lambda_{ik}\phi_i}{1+\lambda_{ik}\phi_i}\right)^{y_{ij}} & \text{if } y_{ij} > 0 \end{cases} \tag{5.10}$$

Hurdle models

As previously said, hurdle models assume that there is only one process by which a zero can be produced and divide the modelling into two steps to introduce extra zeros. The idea is that positive counts occur once a threshold is crossed, or a hurdle is cleared. If the hurdle is not cleared, then we have a count of 0. The first part of the model is typically a binary logistic regression that models whether an observation takes a positive count or not, often taking advantage of the inclusion of the effects of covariates on the probability of an observation being zero. If the value is positive, the "hurdle is crossed" and the conditional distribution of the positive values is governed by a zero-truncated count model, usually Poisson distribution for PH model [99] or truncated negative binomial model for NBH model.
If we denote with ω_i the probability of failure in clearing the “hurdle” and generating a non-zero count and if the truncated Poisson distribution is used then the distribution of Y_{ij} can be written as

$$P_{PH}(y_{ij}|\lambda_{ik}, \omega_i) = \begin{cases} \omega_i & \text{if } y_{ij} = 0 \\ (1-\omega_i)\frac{{\lambda_{ik}}^{y_{ij}} e^{(-\lambda_{ik})}}{y_{ij}!(1-e^{-\lambda_{ik}})} & \text{if } y_{ij} > 0 \end{cases} \tag{5.11}$$

while if the truncated NB distribution is used the model becomes

$$P_{NBH}(y_{ij}|\lambda_{ik},\omega_i) = \begin{cases} \omega_i & \text{if } y_{ij}=0 \\ (1-\omega_i)\dfrac{\frac{\Gamma(y_{ij}+\frac{1}{\phi_i})}{\Gamma(y_{ij}+1)\Gamma(\frac{1}{\phi_i})}\left(\frac{1}{1+\mu_{ik}\phi_i}\right)^{\frac{1}{\phi_i}}\left(\frac{\mu_{ik}\phi_i}{1+\mu_{ik}\phi_i}\right)^{y_{ij}}}{1-\left(\frac{1}{1+\mu_{ik}\phi_i}\right)^{\frac{1}{\phi_i}}} & \text{if } y_{ij}>0 \end{cases} \tag{5.12}$$

It is noteworthy that PH can be seen as a reparameterization of a ZIP, with $\omega_i = \pi_i + (1-\pi_i)e^{-\lambda_{ik}}$. However, in regression contexts different parameters (ω_i or π_i) are modelled and the hurdle and zero-inflated models are no longer equivalent, because in the first case the regression estimates refer to the covariate effects on the log-odds of a zero response, while in the zero-inflated framework they refer to the covariate effects on the log-odds of structural zeros.

5.1.4 metaSPARSim modelling: accounting for extra variance, severe sparsity and feature dependence

The above count modelling procedures that allow for extra variability and extra zeros were developed and are currently used to model count data coming from economics, sociology, geology and many other fields, but they are rarely chosen to simulate count data coming from 16S rRNA gene sequencing experiments. In fact, no simulator in amplicon sequencing count data framework has been published to this purpose that incorporates such models. This may be primarily linked to the need for modelling of some prior information on, e.g., the probability of having a zero count, the knowledge or estimate of which is severely complicated in metagenomic studies by the huge amount of (biological and technical) confounding factors. In addition, the above models consider all the features present in a sample as being independent one from each other. In microbiome count data, a complex microbial population is massively sequenced, thus implying each count being dependent from the level of presence of other taxa in the sample.

The main topic of this thesis is to examine 16S count data variability and sparsity and try to find out which pre-processing pipeline is the most accurate in capturing the real nature of this type of data and in recovering information lost in the sequencing process. In order to do this, the ideal synthetic count data generator should consider the compositional nature of this datum and should be able to produce sparsity and variability in a natural and intrinsic way, avoiding tricky and thus possibly imprecise artificial zero count introduction. To do this, a multivariate model with variable internal probabilities is proposed that follows the

rationale of experimental data production and captures directly the mechanisms of structural and random null values.

To properly model the sequencing process, we may think of it as acting on original sequences present in the sequencing platform as a sampling procedure. In fact, we know original sequences are put into the sequencer and washed on a flowcell where several binding sites are present. The probability of a sequence coming from a bacterial agent to be captured by the binding site and then sequenced is dependent on the abundance of that agent within the total population. Additionally, it is important to consider that when a sequence is captured, it is no longer available for other binding sites, configuring the sampling as a sampling without replacement with a limited number of draws. Previously introduced Poisson and Negative Binomial distributions model count data separately for each OTU, and consequently do not take into account internal dependencies introduced by the non-replacement. Under this approach, the most suitable statistical framework for modelling the sequencing process is obtained by adopting a Multivariate Hypergeometric (MHG), with the number of draws represented by the sequencing depth of samples and internal class abundance varying between biological replicates following a gamma distribution. In particular, let then Y_{ij} be the count value for feature i in experimental group k to which sample j belongs. Then we can write:

$$\begin{aligned} \mathbf{Y}_j &\sim MHG(n_j, c, \mathbf{m}_j, N_j), \\ m_{ik} &\sim gamma(b_i, \theta_{ik}). \end{aligned} \tag{5.13}$$

where

- n_j is the library size of sample j
- c is the number of observed features (OTUs)
- $\mathbf{m}_j = (m_{1j}, m_{2j}, \ldots, m_{cj})$ is the vector indicating the number of fragments coming from OTU i, $i \in \{1, 2, \ldots, M\}$ in sample j
- N_j is the real population size for sample j, that is $N_j = \sum_i m_{ij}$
- b_i and θ_{ik} are shape and scale parameters of the gamma distribution modelling the distribution of DNA fragments in OTU i and group k.

Noting that the probability p_{ij} of sampling a read (count) coming from a microbial component of OTU i is equal to the proportion of original reads coming from that OTU, that is $p_{ij} = \frac{m_{ij}}{N_j}$, we can also write:

$$\mathbf{Y}_j \sim MHG(n_j, c, \mathbf{p}_j \cdot N_j, N_j), \tag{5.14}$$

According to this modelling, zero count values rise naturally from the sampling procedure, following the real scenario in which rare OTUs result more frequently than others in extra zero counts because they are the most probable features not read (i.e. sampled) by the sequencer. As a consequence, contrarily from other available simulators, metaSPARSim does not require a zero inflation step.

5.2 The tool

metaSPARSim is a tool written in *R* language with core function implementation in *C++* (http://sysbiobig.dei.unipd.it/?q=Software#metaSPARSim). It generates datasets with a specified input number of groups (e.g. samples coming from different niches) and biological replicates per group, as exemplified in Figure 5.1. For each group, the simulation takes as input one vector of average abundances and one vector of biological variability. Each vector element is related to a specific OTU mean and variance parameters. These parameters can be specified by the user, estimated from real datasets or taken from a set of pre-coded scenarios that are integrated into the simulator. Biological samples are generated using a OTU-specific *gamma* distribution. Then, the sequencing step is reproduced by sampling the wanted number of reads from each biological sample accordingly to a MHG distribution, whose internal probabilities are defined by sample-specific proportional expressions.

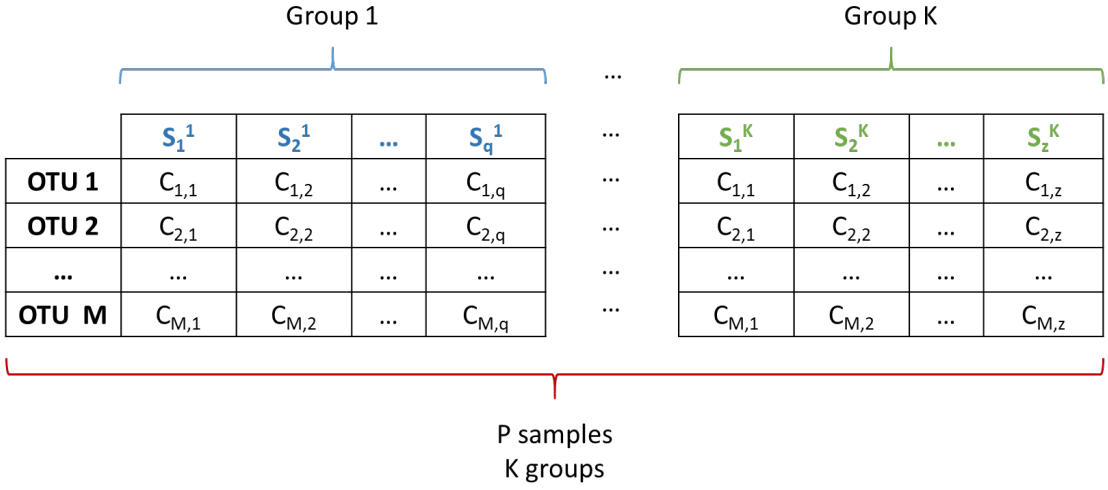


Fig. 5.1 Example of metaSPARSim output matrix. In this example, the matrix corresponds to an experiment in which *K* groups with different number of replicates are simulated. The dataset contains *M* OTUS and a total of *P* samples.

In the following, inputs, outputs and available precoded datasets will be presented.

5.2.1 Inputs

As anticipated, metaSPARSim simulator needs the user to specify only a minimum quantity of parameters. Once the number of desired group to simulate is fixed, the only compulsory parameters to give the simulator are:

- one vector of mean abundances per group
- one vector of OTU variability per group
- one vector of desired library sizes per group, each corresponding to a biological replicate internal to the group

These quantities may be inputed to metaSPARSim in three different ways, as explained in the following.

Case 1. Direct specification

In the first modality, the user can specify his/her own parameters. These may come from a previously performed experiment or could be drawn from theoretical distributions. For example, one may want to test a tool for differences in performance under different conditions, such as varying from low to high dispersed data or changing the underlying distribution that describes the internal subdivision of counts into the OTUs, looking for trends when applying, e.g., a more skewed distribution or a more symmetric one.

Case 2. Estimation procedure from real datasets

If the user have no prior knowledge of count characteristics and his/her aim is not testing for different theoretical scenarios, metaSPARSim also allows for estimation of needed parameters from real count tables. In fact, by the use of built-in functions and given a real count matrix metaSPARSim internally estimates abundances, dispersion and library size vectors for simulation.

Additionally, also the hybrid mode is available. For example, one may want to take information about mean values from a real experiment while using personally specified vectors of dispersions or library sizes. The estimation procedure is indeed composed by single modules, all singularly accessible for specific use. More precisely:

- `estimate_intensity` function: performs the estimate of intensity vectors to give metaSPARSim as input; biological replicates information is summarized calculating their mean for each OTU. In this step, the user can decide whether to consider or discard null values during this procedure.

- `estimate_variability` function: performs the estimate of replicates variability within a group for each OTU calculating the dispersion parameter as in *edgeR* [72].
- `estimate_library_size` function: extracts real library sizes from the original dataset that can be used directly or considered as range for casual new library size extraction.

For complete parameter estimation, function `estimate_parameters_from_data` has been implemented as wrapper of the three above functions.

Case 3. Available presets included in the simulator

Lastly, if no prior information o real datasets are available, the user can simulate his/her own 16S count matrix by taking advantage of the pre-coded scenarios present in the simulator. In fact, different sets of parameters taken from real datasets or synthetically designed to describe theoretical distributions of interest in microbiome studies are currently available and ready to use just selecting them from the package. This allows researchers whose interest is not linked to a specific parameter configuration or real situation to obtain rapidly some datasets on which to test their tools, without loosing time in looking for parameters to work on a plausible and realistic matrix.

After having received the input parameters, metaSPARSim starts to simulate synthetic data by using a gamma distribution with given mean intensity and variability vectors to obtain biological replicates and then simulates the sequencing process through the multivariate hypergeometric distribution, as shown in Figure 5.2.

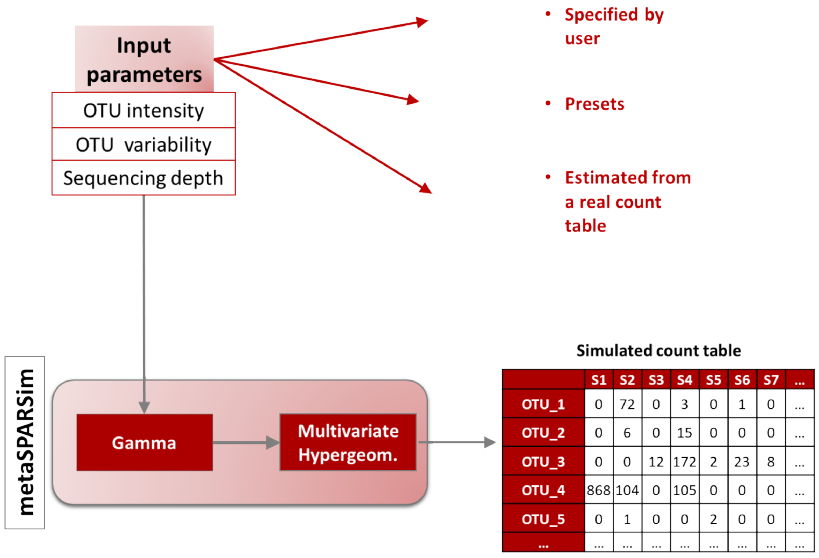


Fig. 5.2 metaSPARSim simulation workflow.

5.2.2 Outputs

After simulation processing, metaSPARSim will output its results as a list composed by the following elements:

- 'counts': a matrix containing the final count matrix for developer's use
- 'gamma': a matrix containing sample abundance values before sequencing step simulation. This should be intended as the "true" values to use as golden standard when testing, for example, normalization tools. In fact, pre-sequencing data and normalized data should have a structure as similar as possible to assure sequencing artifacts are corrected before downstream analyses.
- 'params': a collection of the complete set of parameters used for simulation.

5.2.3 Presets

As previously said, a set of pre-computed parameters for 16S count matrix simulation was included in metaSPARSim. The available presets were obtained both as estimate from real datasets ("real data presets" in the following) and as modelling of theoretical scenarios described by statistical distributions ("synthetic presets").

Real data presets. The parameter sets collected in this Section have been estimated using metaSPARSim from five different real data, which details are reported in in Table

5.1. These datasets were chosen to represent 16S rDNA sequencing data diversity, having all of them peculiar characteristics in terms of number of samples, constitutive groups, sequencing depths and sample source. The first two sets were estimated from data coming from experiments performed in the context of this work, while the other ones are estimated from Human Microbiome Project (HMP) [90, 89] data (*R3-R4* sets) and Atacama soil [100] data (*R5*).

Synthetic presets. These parameters sets were included in metaSPARSim for developers usage for testing their tools when hypothesising a theoretical microbiome internal distribution. For example, the first synthetic set (*S1*, Table 5.2) was produced to simulate ecosystems in which species are uniformly distributed into the samples. The second preset, *S2*, was simulated to provide developers a theoretical situation in which species are unevenly distributed into the samples, with the great majority of species having low-abundances and a few species highly represented in the samples. The last set, was created to reproduce ecosystems in which species have normally distributed abundances around a mean.

Table 5.1 metaSPARSim presets estimated from real data.

Name	Source	Groups	Samples	Replicates per group	Features	Sequencing depth (range)
R1	Chicken caecum	22	110	5	3541	88692-832309
R2	Raw milk cheese	40	118	2-3	3109	28536-394754
R3	Human body	8	40	5	758	2798-24095
R4	Human body	8	800	100	1767	1010-49414
R5	Soil	11	39	2-5	5489	3158-98639

Table 5.2 metaSPARSim presets based on theoretical scenarios described by statistical distributions.

Name	Distribution	Groups	Samples	Replicates per group	Features	Sequencing depth (range)
S1	Uniform	15	150	10	5000	$10^3 - 10^5$
S2	Weibull	15	150	10	5000	$10^3 - 10^5$
S3	Normal	15	150	10	5000	$10^3 - 10^5$

5.3 Evaluation criteria

To explore metaSPARSim performance on simulating realistic datasets, a comparison between the original data and the simulated ones was performed. This comparison is based on the factors that most characterize and define sequencing count data, as follows.

- Sparsity: the number of zero counts over the total number of entries is calculated, as well as the level of sparsity (expressed as percentage) per row and per column
- Intensity: count values intensity and their distribution within the sample

- Variability: expressed both as variance and relative variance, measures the dispersion of data among replicates.

The goodness in reproducing realistic characteristics was evaluated by both qualitative and quantitative means:

- **Q–Q (quantile-quantile) plots** [101]: a graphical method for comparing two probability distributions by plotting their quantiles against each other. The two compared distributions are considered to be equal when plotted data lay on the diagonal.
- **boxplots** [102]: a method for graphically representing numerical data distributions through their quartiles.
- **RDI (Raw data, Descriptive statistics, and Inferential statistics) plots** [103]: this kind of plots permit to represent both punctual and distribution information, joining scatter plot, box plot and density plot together.
- **Mann-Whitney U test** [104]: applied to relative abundances vectors. This test is based on the null hypothesis that the two tested samples come from the same population (i.e. they both have the same median). If the resultant *P*-value is less than a fixed significance level (here 0.05), then the null hypothesis is rejected in favour of the alternative hypothesis, i.e. the two samples come from different populations. This test was used to check for possible statistically significant dissimilarities between real and simulated vectors of intensities and variances.
- **Cohen's *d*** [105]: applied to relative abundances vectors. It is one measure associated with the calculation of so-called effect size, a quantitative measure of the magnitude of a phenomenon; here, the effect size quantifies the size of the difference between two groups. In this analysis, we decided to include it alongside the significance test because it has been widely shown ([106],[107],[108]) that when examining effects using large samples significant testing can be misleading because even small or trivial effects are likely to produce statistically significant results. Thus, reporting only the significant P-value for an analysis is not adequate to fully understand the results. Cohen's d is defined as the difference between two means divided by a standard deviation for the data, i.e.

$$d = \frac{\mu_1 - \mu_2}{s}, \tag{5.15}$$

where s is the pooled standard deviation, defined as

$$s = \sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2}} \tag{5.16}$$

where n_1 and n_2 are the two sample sizes and s_1 and s_2 are the related standard deviations. Table 5.3 contains cut-offs for magnitudes interpretation, as initially suggested by Cohen and expanded by Sawilowsky[109].

Table 5.3 Cohen's *d* cut-offs.

Effect size	*d*	Reference
Negligible	< 0.01	Sawilowsky, 2009
Very small	0.01	Sawilowsky, 2009
Small	0.20	Cohen, 1988
Medium	0.50	Cohen, 1988
Large	0.80	Cohen, 1988
Very large	1.20	Sawilowsky, 2009
Huge	2.0	Sawilowsky, 2009

Chapter 6

metaSPARSim16S performance assessment

As previously introduced, the goodness of metaSPARSim simulations was assessed using both publicly available datasets, such as Human Microbiome Project (HMP) data [89, 90], and self-produced data. These datasets were chosen to obtain a range of scenarios as wide as possible, including experiments based on different sequencing platforms and with diverse group, sample and replicate numbers and sequencing depths. This allowed to obtain a solid assessment of performance in different situations, the results of which are shown in the following Sections. Results will be shown grouped by the three above mentioned main characterizing aspects of 16S count data matrices, i.e. sparsity, count intensity and count variability, jointly for animal gut, raw milk cheese and HMP data. In this section, results on the HMP dataset obtained sampling 5 samples from 8 chosen groups considered in HMP study are shown. Results from the bigger dataset (100 samples per group) are here omitted for brevity and analogy with the other dataset and reported in Appendix A.

6.1 Sparsity

The first examined metric was the level of sparsity of the simulated matrix. The overall zero abundances was very well reproduced in all the datasets, the real ones being of 78.7%, 97.1% and 81.4% and the simulated ones of 77.9%, 93.8% and 80% respectively for animal gut, raw milk cheese and HMP data. For all the three datasets, in Figure 6.1 the true sparsity percentages calculated on group submatrices is plotted against the simulated ones. As can be seen, all the results confirm that the accuracy in recreating datasets with realistic overall sparsity remains valid when looking at intra-group sparsity. Additionally, zeros-by-

row (feature) and zeros-by-column (samples) distributions were calculated to check if the simulator is able to reconstruct not only the true abundance but also the true location of zero counts. As showed in Figure 6.2, the zero distributions per row and per column are well reproduced in all the datasets, the worst performance being observed in food microbiota dataset. We recall that this is the more challenging dataset of the chosen ones, due both to the dramatic total sparsity level and to the lower number of available replicates (2/3). Nevertheless, the results remained quite good, the Q-Q plot lying on the diagonal for the great majority of values.

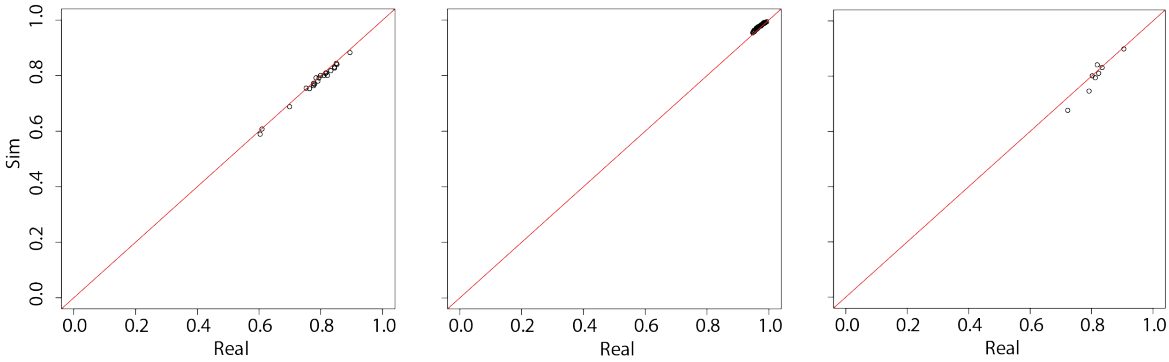


Fig. 6.1 Scatter plot of group-specific percentage of zeros in real and simulated datasets. From the left, animal gut, raw milk cheese and HMP data results.

6.2 Intensity

Regarding the intensity of count values, metaSPARSim was able to capture the nature of non-zero values both globally and specifically for each group. Count values in real and simulated data showed very similar characteristics, both looking at the overall distribution of mean values (Figure 6.3 (A, C, E)) and at the punctual information (Figure 6.3 (B, D, F)). Again, the best performance was achieved for animal gut microbiota dataset, while the worst was obtained for raw milk cheese data where the above introduced critical issues influenced also this metric. Results were, however, still sufficiently accurate to assure a realistic reproduction of a food microbiome experiment, the median and inter-quartile ranges of intensity distributions being realistic values.

As showed in Figures 6.4-6.6, Figures 6.7-6.9 and Figures 6.10-6.12, the performance observed for whole count data were maintained when looking at the replicate means within each group, assuring that the overall good behaviour was a result of a detailed good estimation and not a mean of over and under estimation between biological conditions. Also in this case,

the reduced number of replicates of raw milk cheese dataset led to the worst performance, that remained however acceptable in median for almost all the groups.

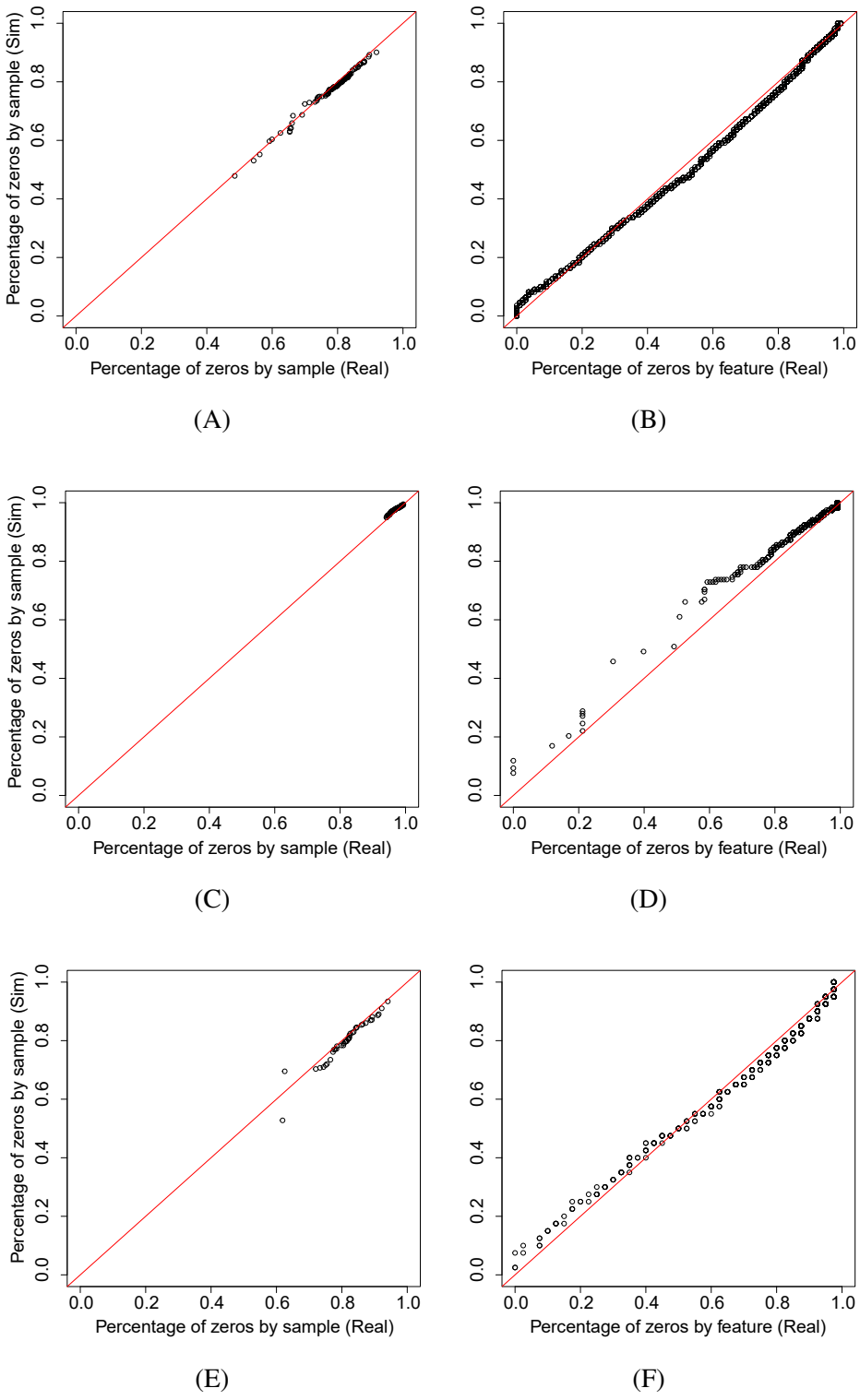


Fig. 6.2 Q−Q plot of percentage of zeros in real and simulated datasets, calulated by sample (A, C, E) and by feature (B, D, F) for animal gut (first row), raw milk cheese (second row) and HMP (third row) data

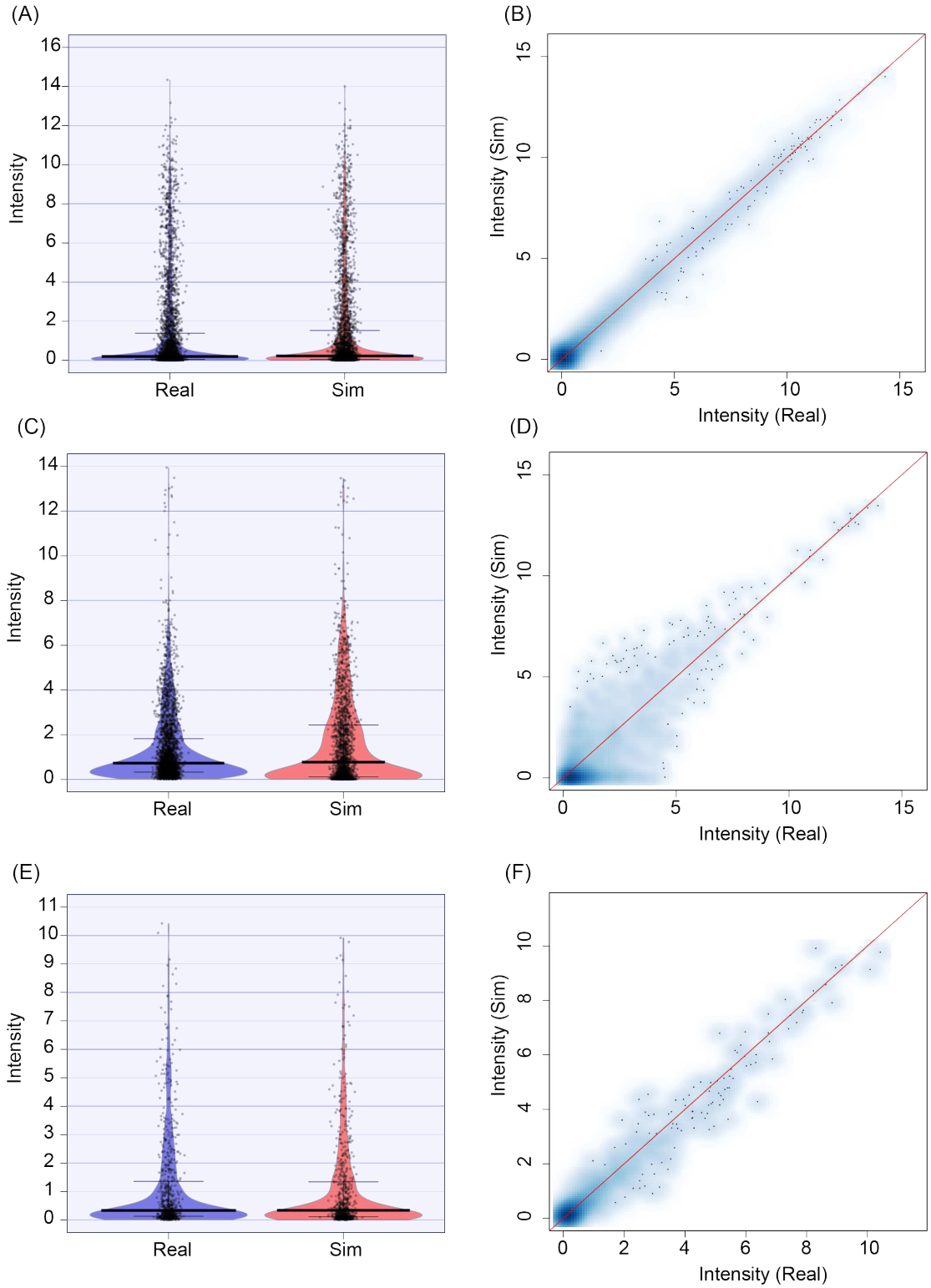


Fig. 6.3 Comparison of Log2 count intensity in real and simulated datasets, represented as RDI plot (A, C, E) and scatter plot (B, D, F) for animal gut (first row), raw milk cheese (second row) and HMP (third row) data, excluding zero mean features.

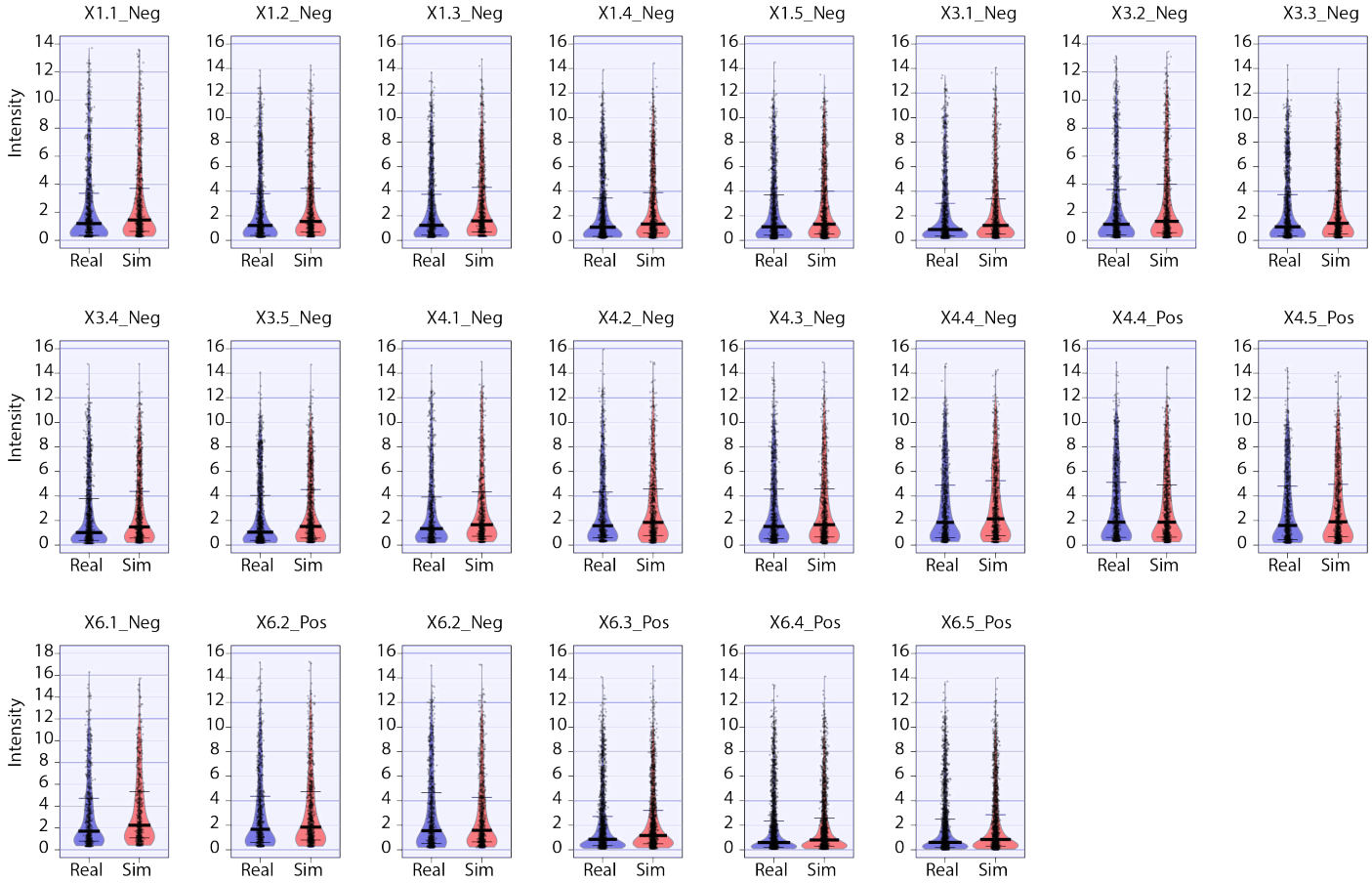


Fig. 6.4 RDI plots of Log2 count values of real and simulated data within each group for animal gut dataset, excluding zero mean features.

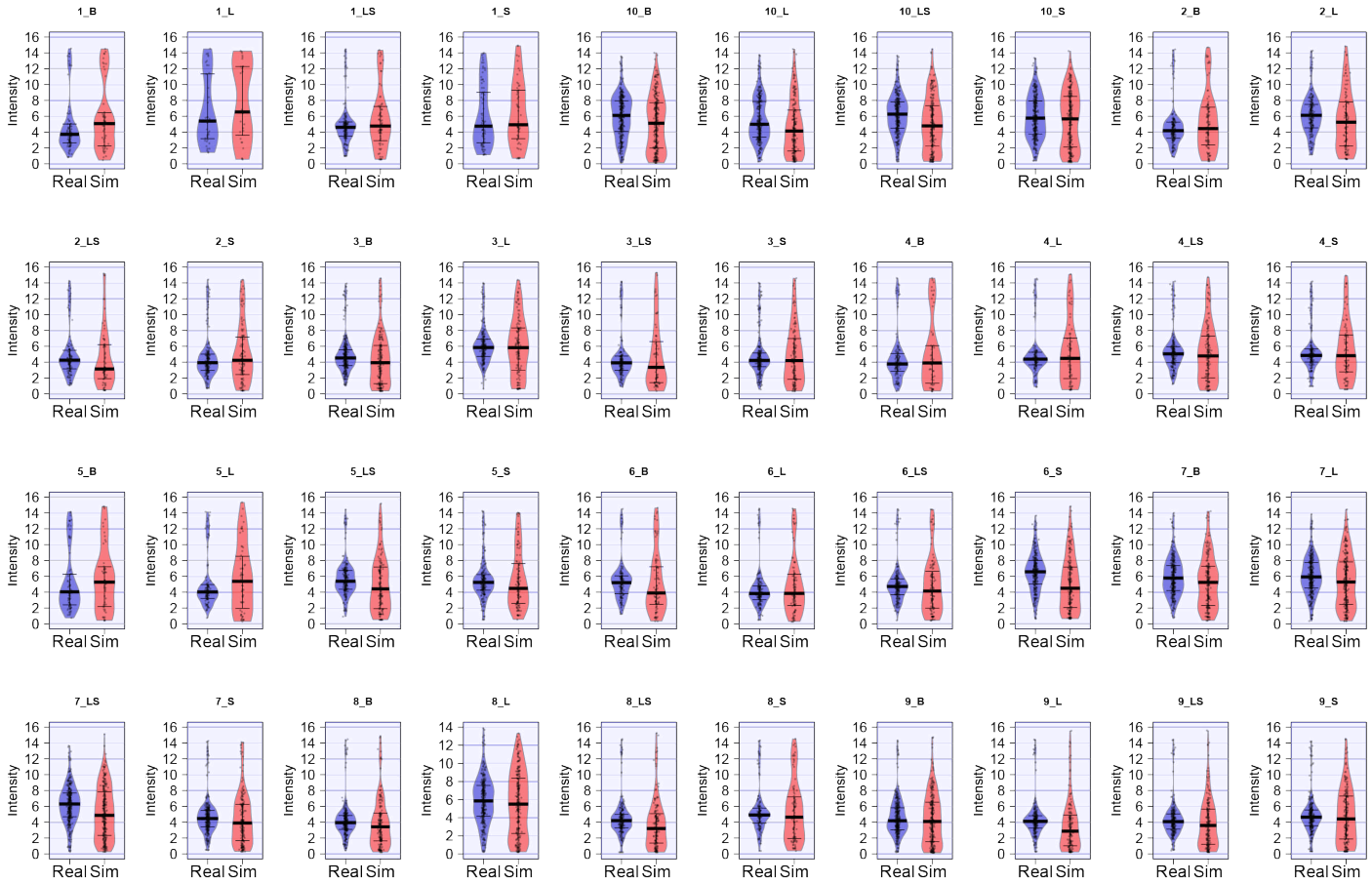
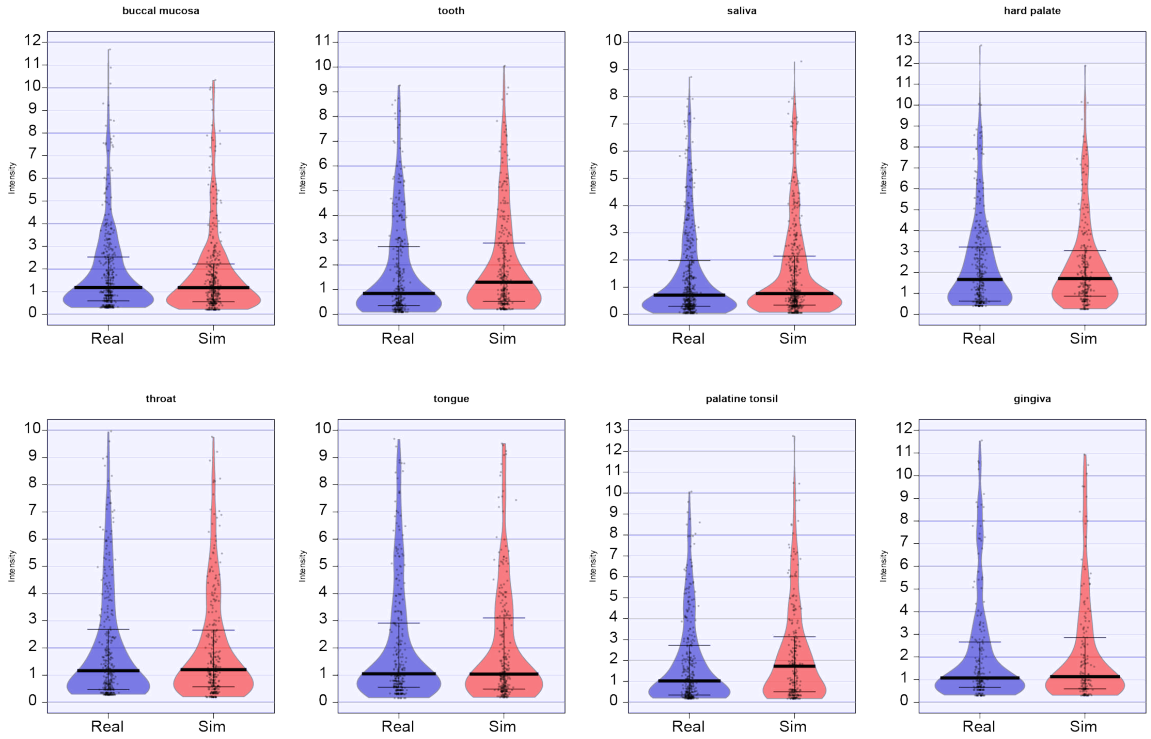


Fig. 6.5 RDI plots of Log2 count values of real and simulated data within each group for raw milk cheese dataset, excluding zero mean features.



Lorem ipsum

Fig. 6.6 RDI plots of Log2 count values of real and simulated data within each group for HMP dataset, excluding zero mean features.

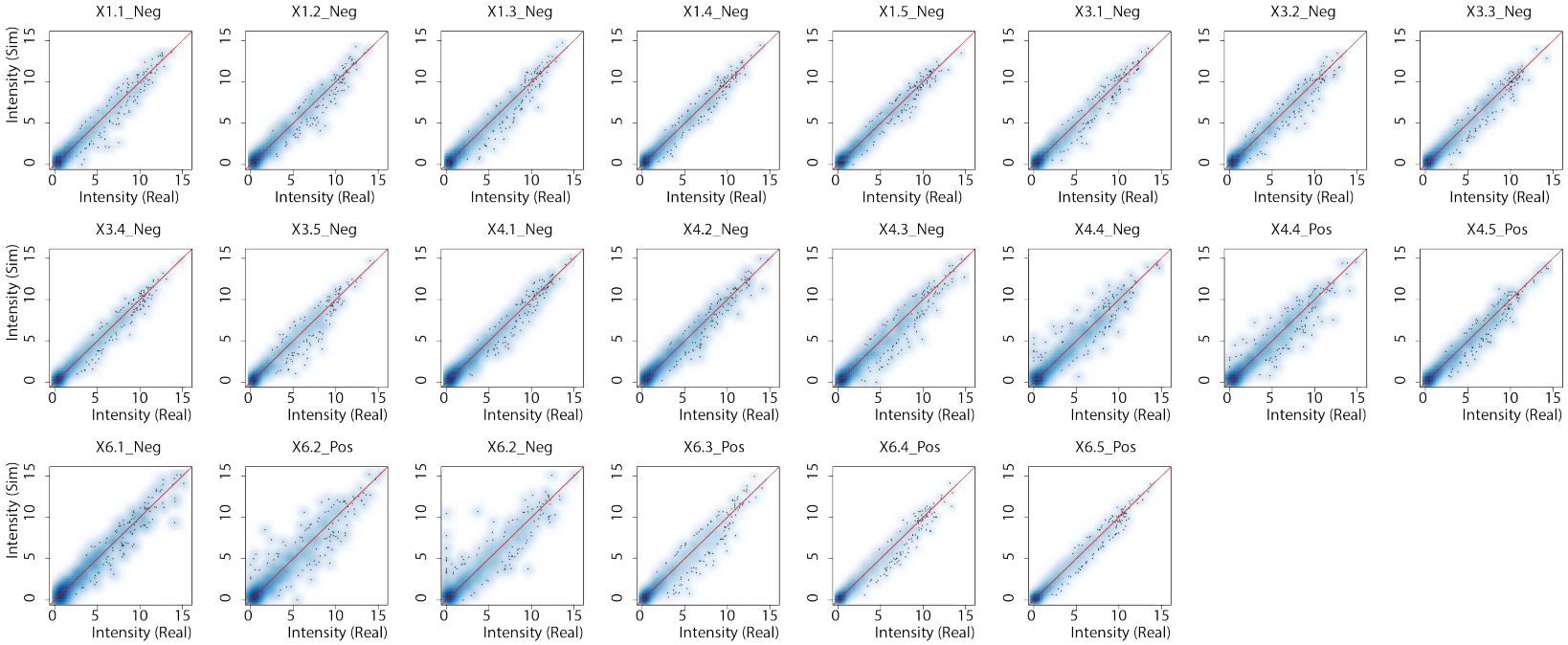


Fig. 6.7 Scatter plots of Log2 count values of real and simulated data within each group for animal gut dataset, excluding cases with zero counts in both data.

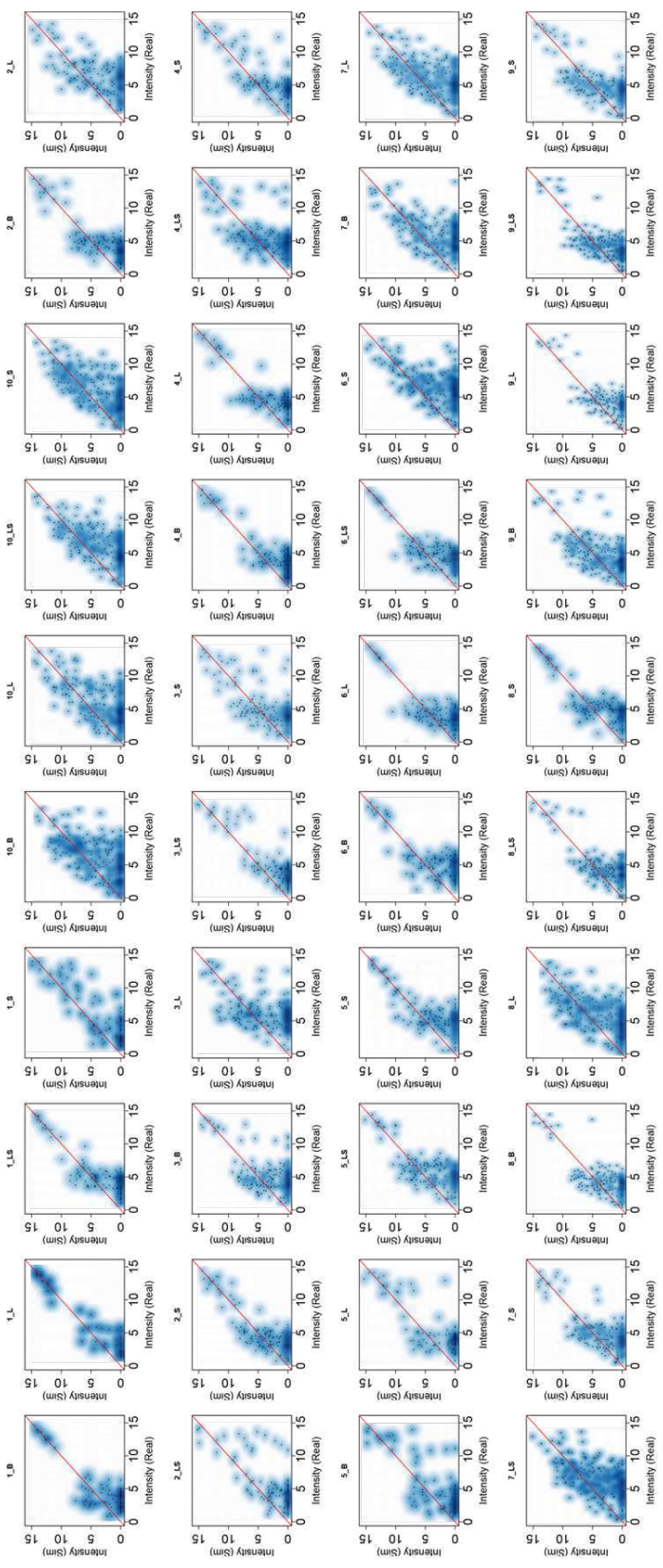


Fig. 6.8 Scatter plots of Log2 count values of real and simulated data within each group for raw milk cheese dataset, excluding cases with zero counts in both data.

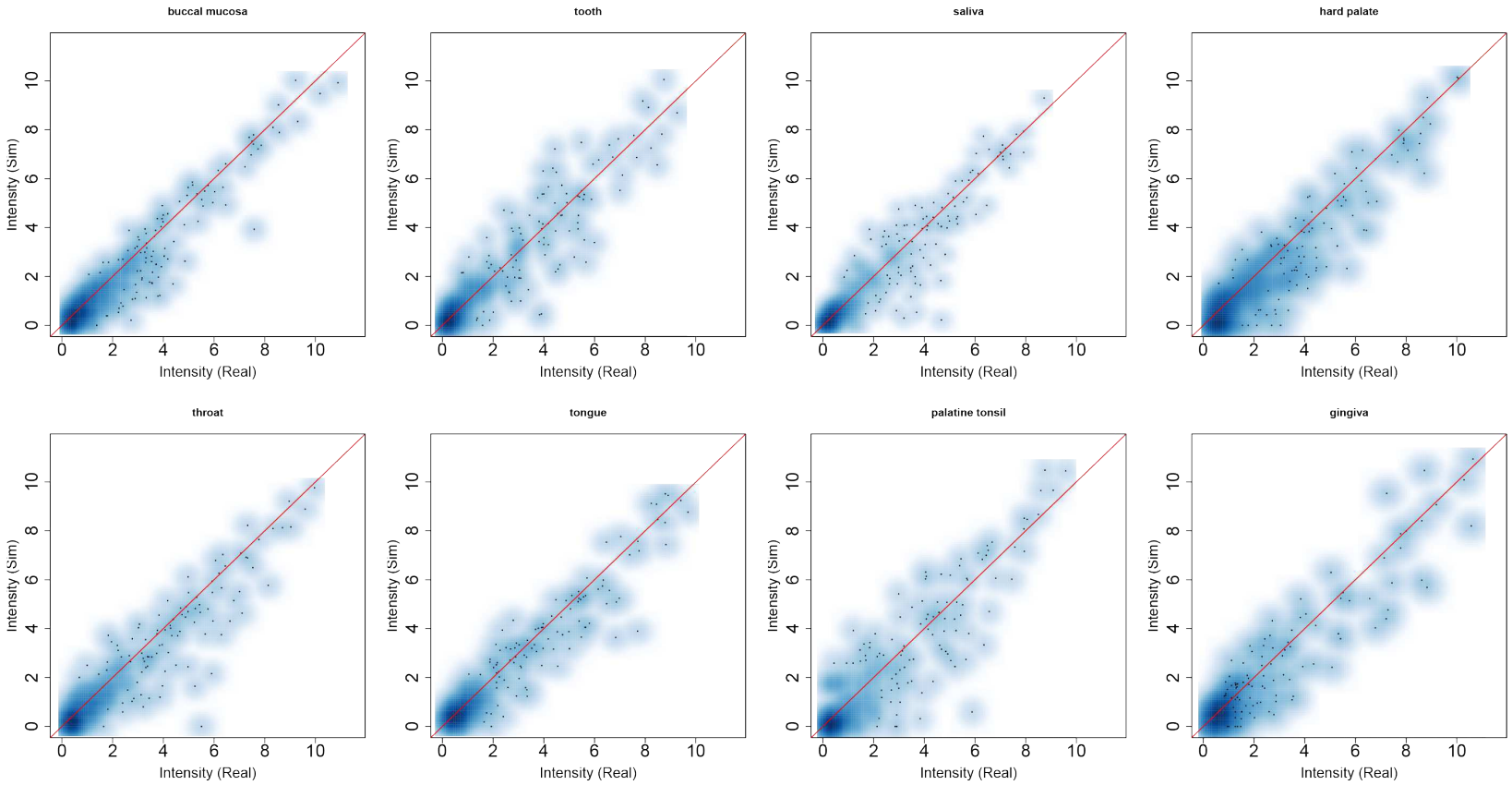


Fig. 6.9 Scatter plots of Log2 count values of real and simulated data within each group for HMP dataset, excluding cases with zero counts in both data.

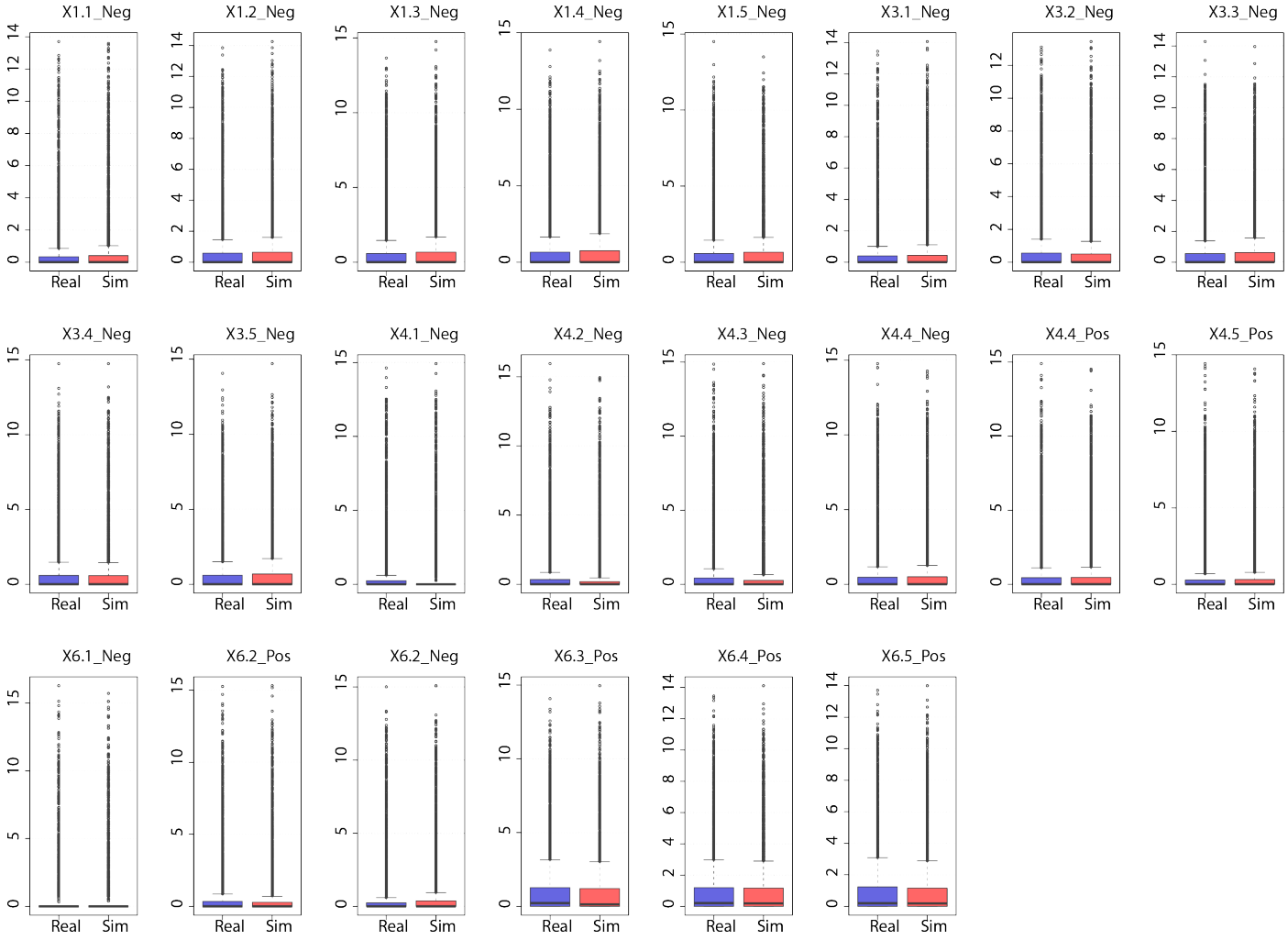


Fig. 6.10 Box plots of Log2 count values of real and simulated data within each group for animal gut dataset.

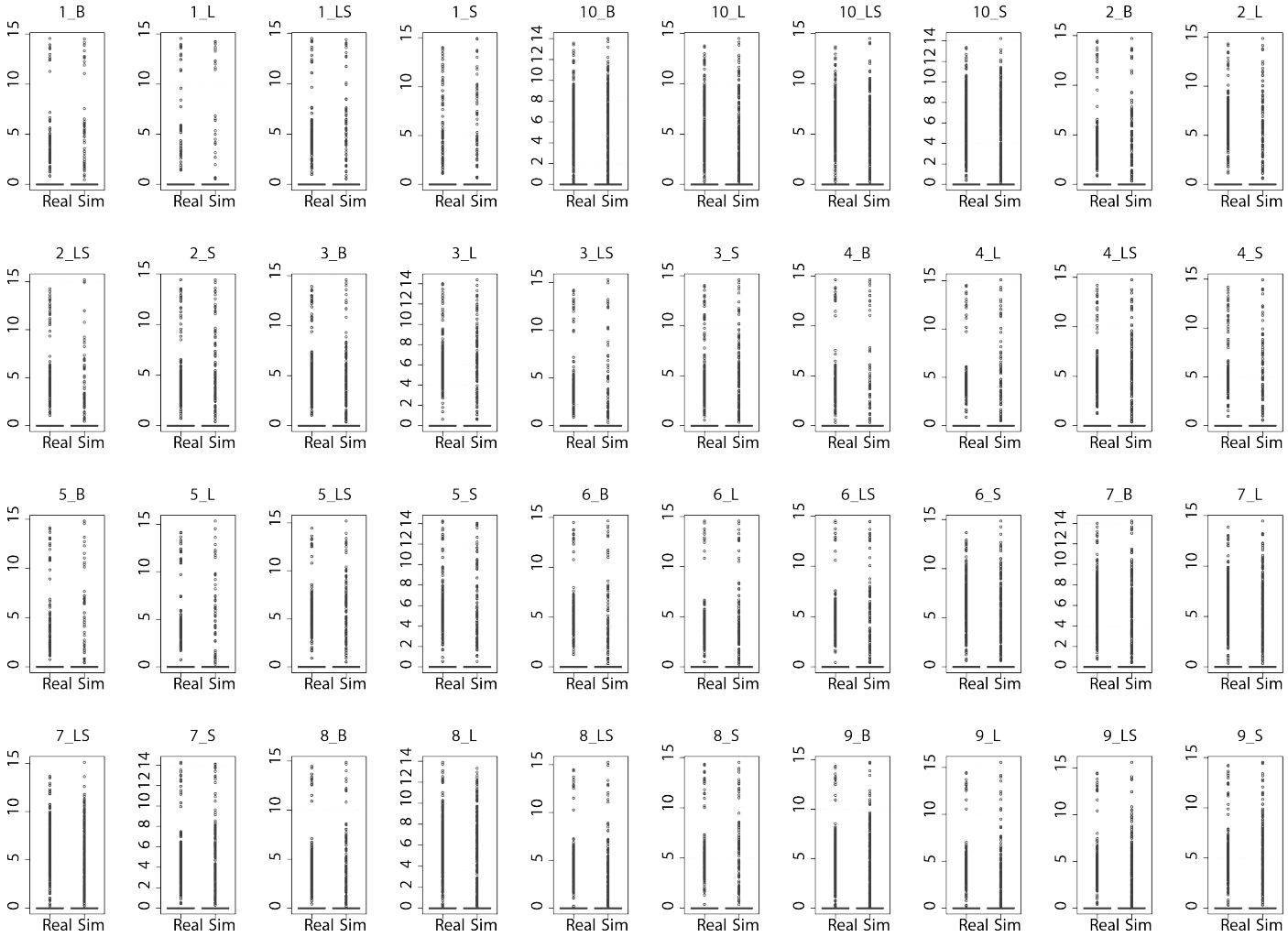


Fig. 6.11 Box plots of Log2 count values of real and simulated data within each group for raw milk cheese dataset.

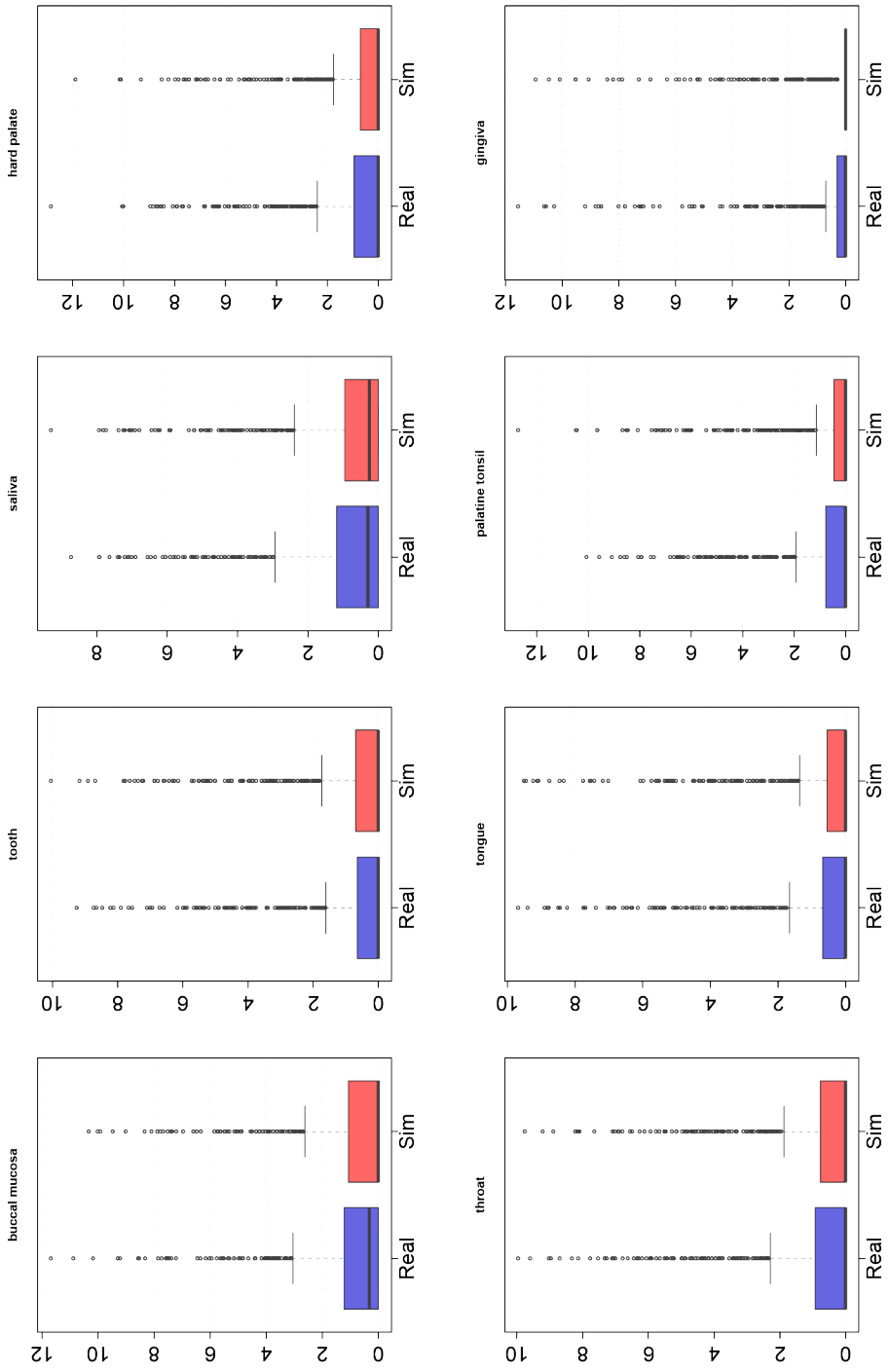


Fig. 6.12 Box plots of Log2 count values of real and simulated data within each group for HMP dataset.

Moreover, the mean between replicates (normalized data) for each feature was computed and a Mann-Whitney U test was performed to test for group mean distribution differences between real and simulated data for each dataset. Additionally, the effect size was calculated and a bootstrap procedure (10000 draws with 5% of total feature number) paired with a Mann-Whitney U test was performed to test for significance in subsamplings, thus overcoming sampling size issue for significance. Results (shown in Tables 6.1,6.2,6.3) confirmed real and simulated values came from the same distribution. In fact, although many groups showed significant differences for the global test, the related effect size was always found to be negligible and the percentage of bootstrap extractions in which significance was found was null for all the groups, thus confirming significance was strongly due to the huge sample sizes (number of features). As expected, bootstrap results showed a slight increase in percentage of significant tests for raw milk cheese dataset compared to other datasets, even though it always remained very low (maximum value: 8.89%).

Table 6.1 Mann-Whitney U tests and effect size results in comparing real and simulated mean count distribution within groups for animal gut dataset.

Group	***P* value**	**Significance**	**Cohen's *d* magnitude**	**Bootstrap significance (%)**
1	0.039	Y	Negligible	0
2	0.027	Y	Negligible	0
3	0.001	Y	Negligible	0
4	0.013	Y	Negligible	0
5	0.109	N	– – –	0
6	0.002	Y	Negligible	0
7	0.013	Y	Negligible	0
8	0.013	Y	Negligible	0
9	0.002	Y	Negligible	0
10	0.004	Y	Negligible	0
11	0.009	Y	Negligible	0
12	0.133	N	– – –	0
13	0.02	Y	Negligible	0
14	0.373	N	– – –	0
15	0.757	N	– – –	0
16	0.065	N	– – –	0
17	0.020	Y	Negligible	0
18	0.194	N	– – –	0
19	0.179	N	– – –	0
20	0.001	Y	Negligible	0
21	0.006	Y	Negligible	0
22	0.002	Y	Negligible	0

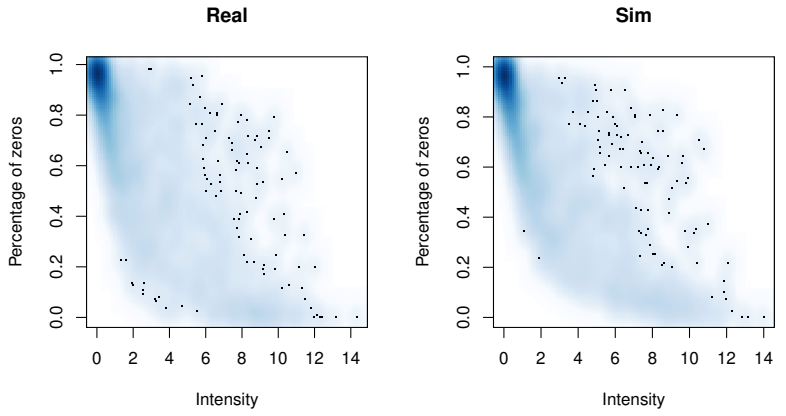
Table 6.2 Mann-Whitney U tests and effect size results in comparing real and simulated mean count distribution within groups for raw milk cheese dataset.

Group	*P* value	Significance	Cohen's *d* magnitude	Bootstrap significance (%)
1	0.004	Y	Negligible	0.88
2	0.111	N	– – –	0.10
3	0	Y	Negligible	5.36
4	0.008	Y	Negligible	0.48
5	0.001	Y	Negligible	0.20
6	0	Y	Negligible	1.96
7	0	Y	Negligible	1.09
8	0	Y	Negligible	0.58
9	0	Y	Negligible	2.34
10	0	Y	Negligible	6.32
11	0	Y	Negligible	3.86
12	0	Y	Negligible	1.35
13	0	Y	Negligible	8.89
14	0	Y	Negligible	2.99
15	0	Y	Negligible	8.01
16	0	Y	Negligible	4.33
17	0	Y	Negligible	5.38
18	0.003	Y	Negligible	0.53
19	0.002	Y	Negligible	0.35
20	0	Y	Negligible	3.58
21	0.005	Y	Negligible	0.83
22	0	Y	Negligible	1.94
23	0	Y	Negligible	1.29
24	0	Y	Negligible	4.00
25	0	Y	Negligible	1.81
26	0	Y	Negligible	2.60
27	0	Y	Negligible	2.15
28	0	Y	Negligible	2.02
29	0	Y	Negligible	7.15
30	0	Y	Negligible	3.99
31	0.001	Y	Negligible	0.19
32	0	Y	Negligible	1.97
33	0	Y	Negligible	4.69
34	0	Y	Negligible	0.61
35	0	Y	Negligible	0.91
36	0.002	Y	Negligible	0.60
37	0	Y	Negligible	0.51
38	0	Y	Negligible	4.28
39	0.002	Y	Negligible	0.16
40	0	Y	Negligible	0.90

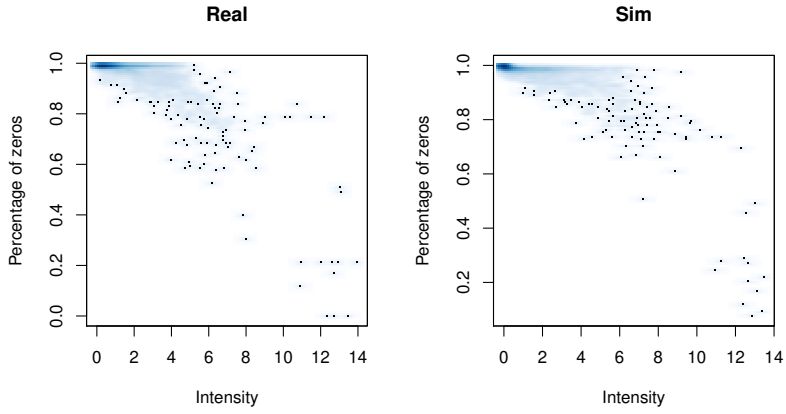
Table 6.3 Mann-Whitney U tests and effect size results in comparing real and simulated mean count distribution within groups for HMP dataset.

Group	*P* value	Significance	Cohen's *d* magnitude	Bootstrap significance (%)
1	0.016	Y	Negligible	0
2	0.056	N	– – –	0.01
3	0.052	N	– – –	0
4	0.016	Y	Negligible	0.01
5	0.024	Y	Negligible	0
6	0.228	N	– – –	0
7	0	Y	Negligible	0.63
8	0.141	N	– – –	0

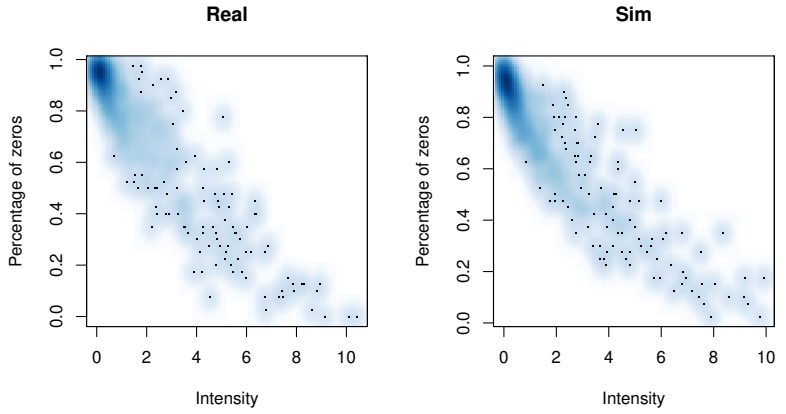
To further investigate the ability to reproduce real data structure and characteristics, the relation between the first two metrics (intensity and sparsity) was studied. As reported in Figure 6.13, the connection between mean intensity and sparsity was accurately maintained from real to simulated datasets, for all the three investigated frameworks.



(A)



(B)



(C)

Fig. 6.13 Scatter plot of the relation between Log2 count mean intensity and sparsity in real and simulated datasets, for animal gut (first row), raw milk cheese (second row) and HMP (third row) data, excluding zero mean features.

6.3 Variability

For variability metric, results are shown in terms of both variance and relative variance (RV, or variance-to-mean ratio), i.e.

$$RV = \frac{\sigma^2}{\mu}. \tag{6.1}$$

As for intensity, metaSPARSim was able to reproduce in a realistic way data variability, both in an overall context (Figure 6.14) and in the detail for each group, as shown in Figures 6.15−6.17, Figures 6.18-6.20, Figures 6.21-6.23 and Figures 6.24-6.26. Also for this metric, raw milk cheese dataset was the most challenging among others. Figures 6.15-6.23 show an underestimation trend especially visible in low variance values, with the distribution of the simulated one being less concentrated around its median. However, metaSPARSim was able to keep an acceptable reproduction of data variability also for this very challenging dataset, as verified by statistical tests and effect size calculation (Tables 6.4-6.6). In fact, no noteworthy difference was found between real and simulated data variance, the effect sizes being always negligible and the bootstrap percentage of significant tests always being under the 9%.

The loss of performance in this and other metrics could be linked with variability estimation complexity, a direct derivation of the very low number of biological replicates, that in some groups are reduced at two. In fact, the less available points for variability estimation, the less is the precision with which the intrinsic nature of data can be caught. This might have caused the overall loss in precision in identifying the internal structure of this dataset.

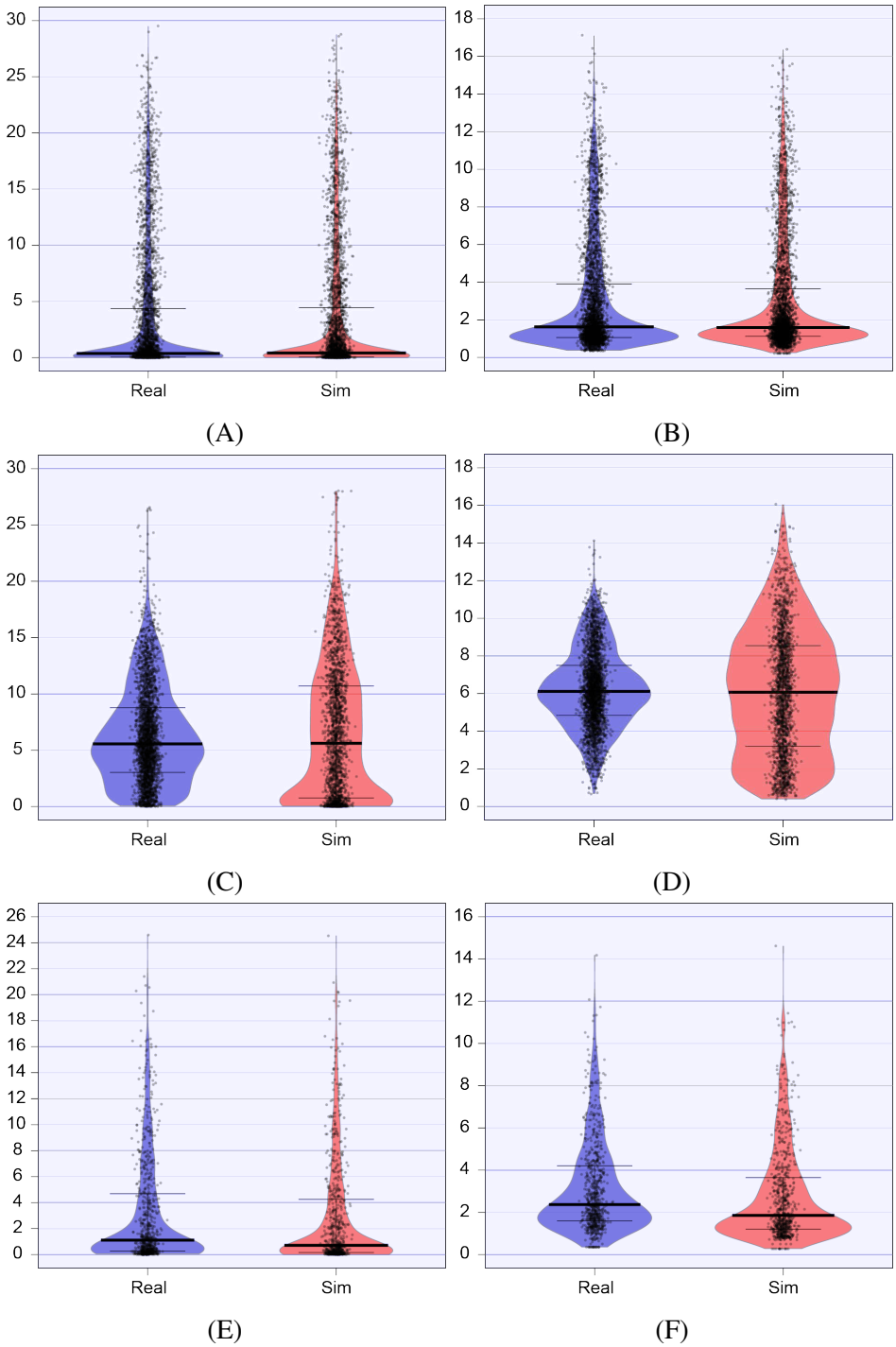


Fig. 6.14 Comparison of Log2 variability values in real and simulated datasets, calculated as variance (A, C, E) and RV (B, D, F)., Results are shown for animal gut (first row), raw milk cheese (second row) and HMP (third row) data, excluding zero mean features.

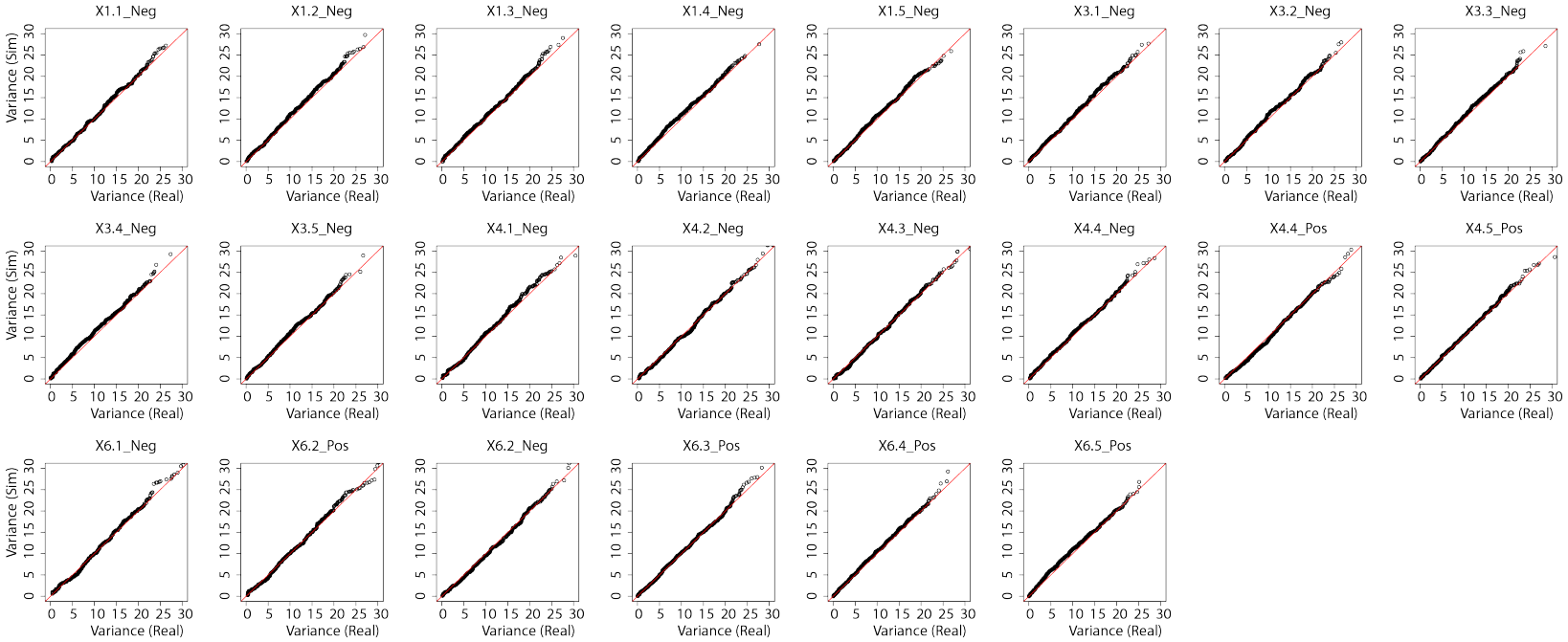


Fig. 6.15 Q−Q plots of of Log2 variance values in real and simulated datasets within each group for animal gut dataset, excluding zero mean features.

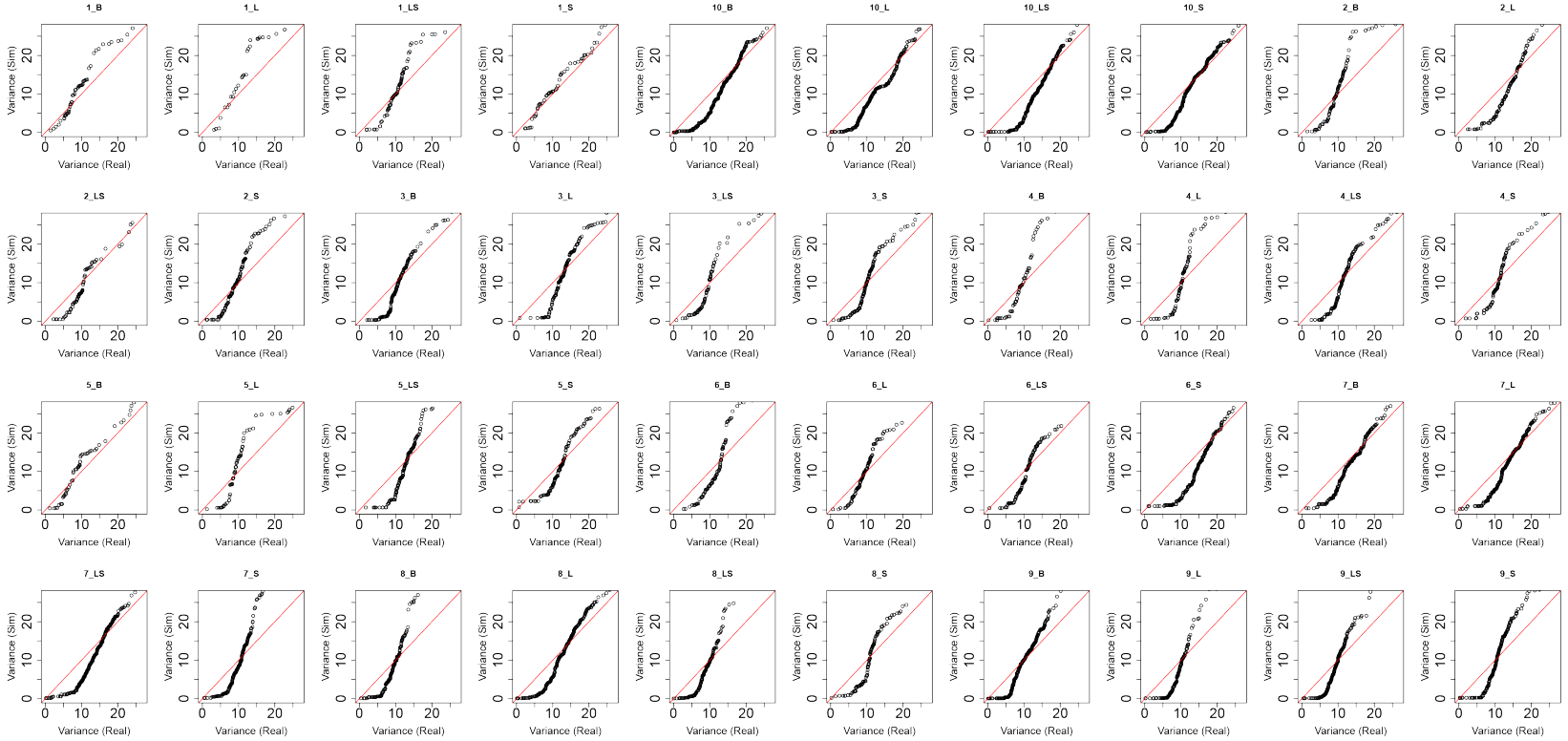


Fig. 6.16 Q−Q plots of of Log2 variance values in real and simulated datasets within each group for raw milk cheese dataset, excluding zero mean features.

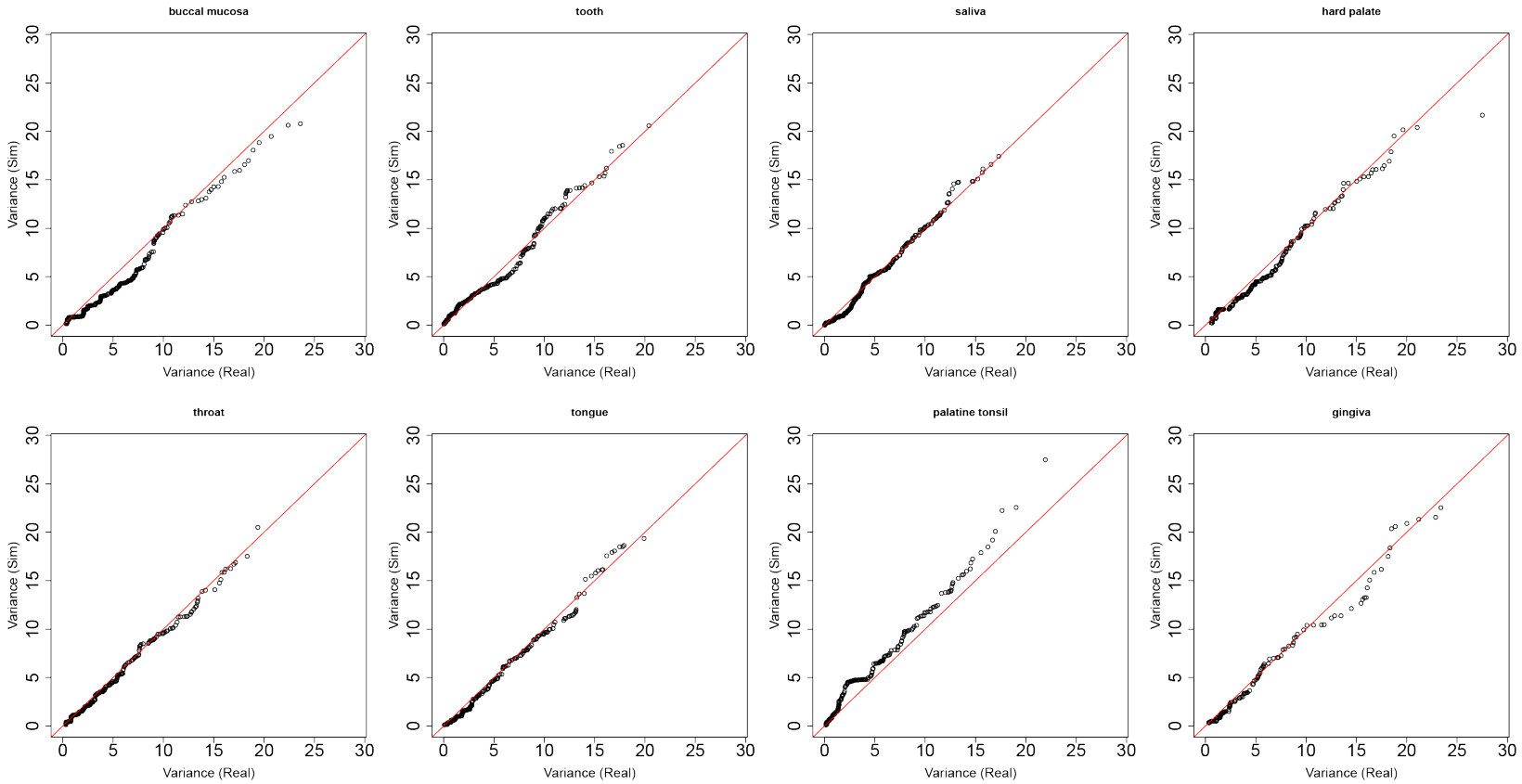


Fig. 6.17 Q−Q plots of of Log2 variance values in real and simulated datasets within each group for HMP dataset, excluding zero mean features.

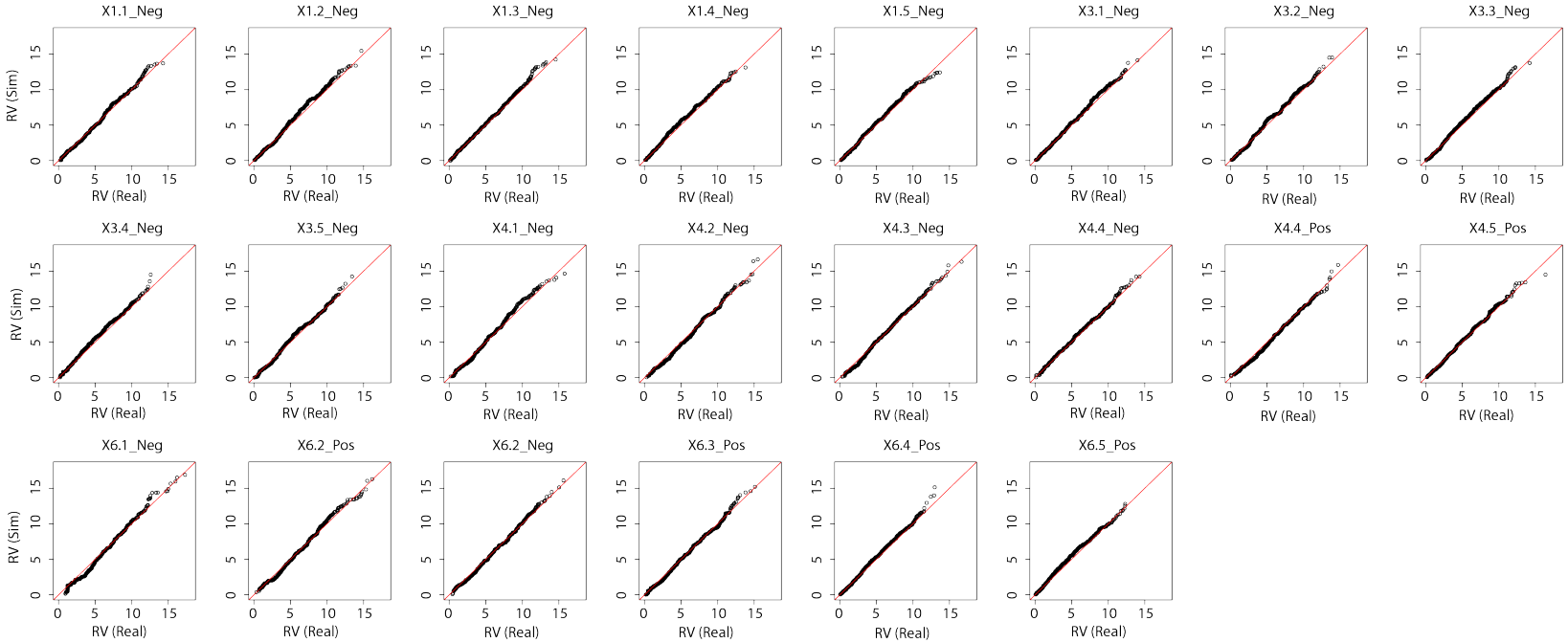


Fig. 6.18 Q−Q plots of of Log2 RV values in real and simulated datasets within each group for animal gut dataset, excluding zero mean features.

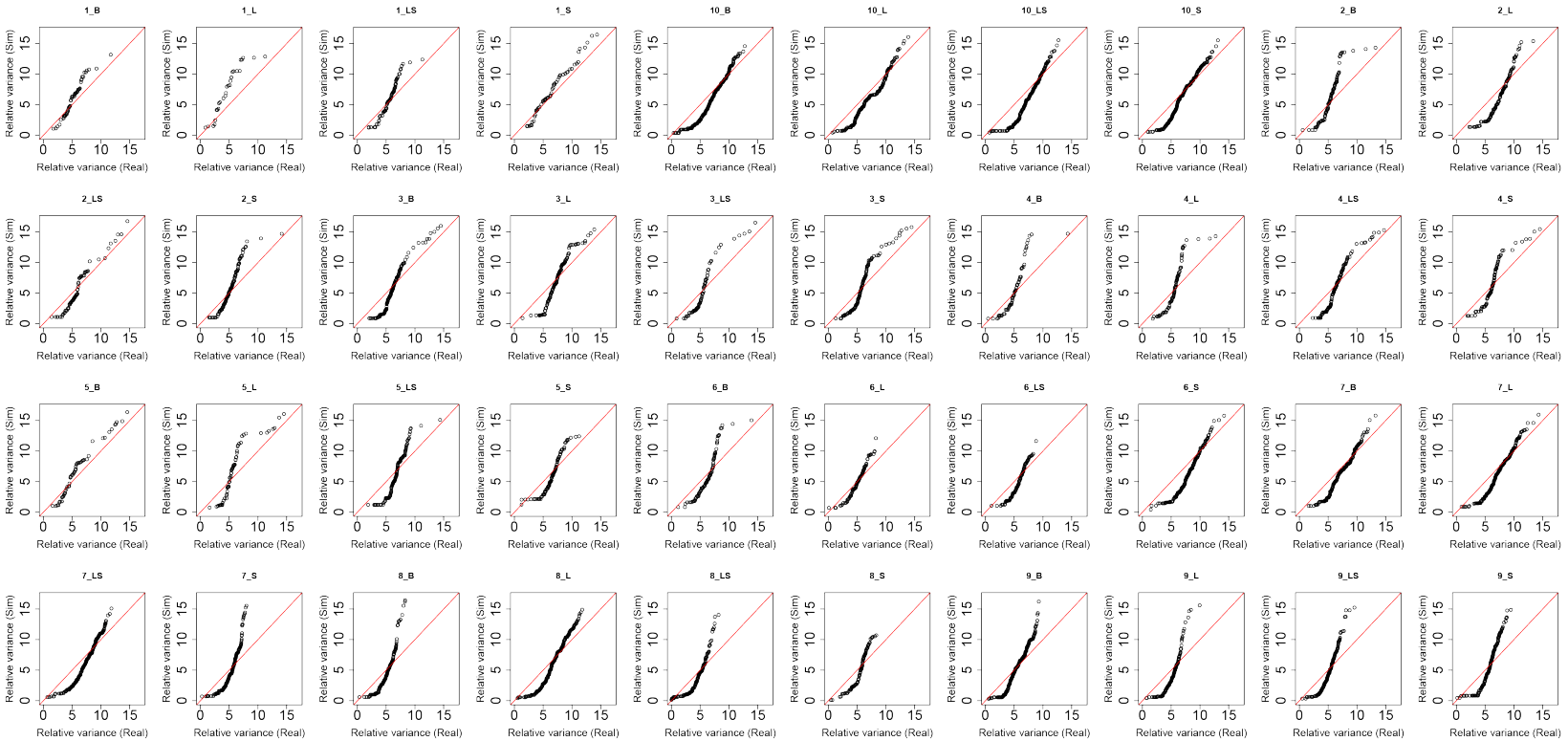


Fig. 6.19 Q−Q plots of of Log2 RV values in real and simulated datasets within each group for raw milk cheese dataset, excluding zero mean features.

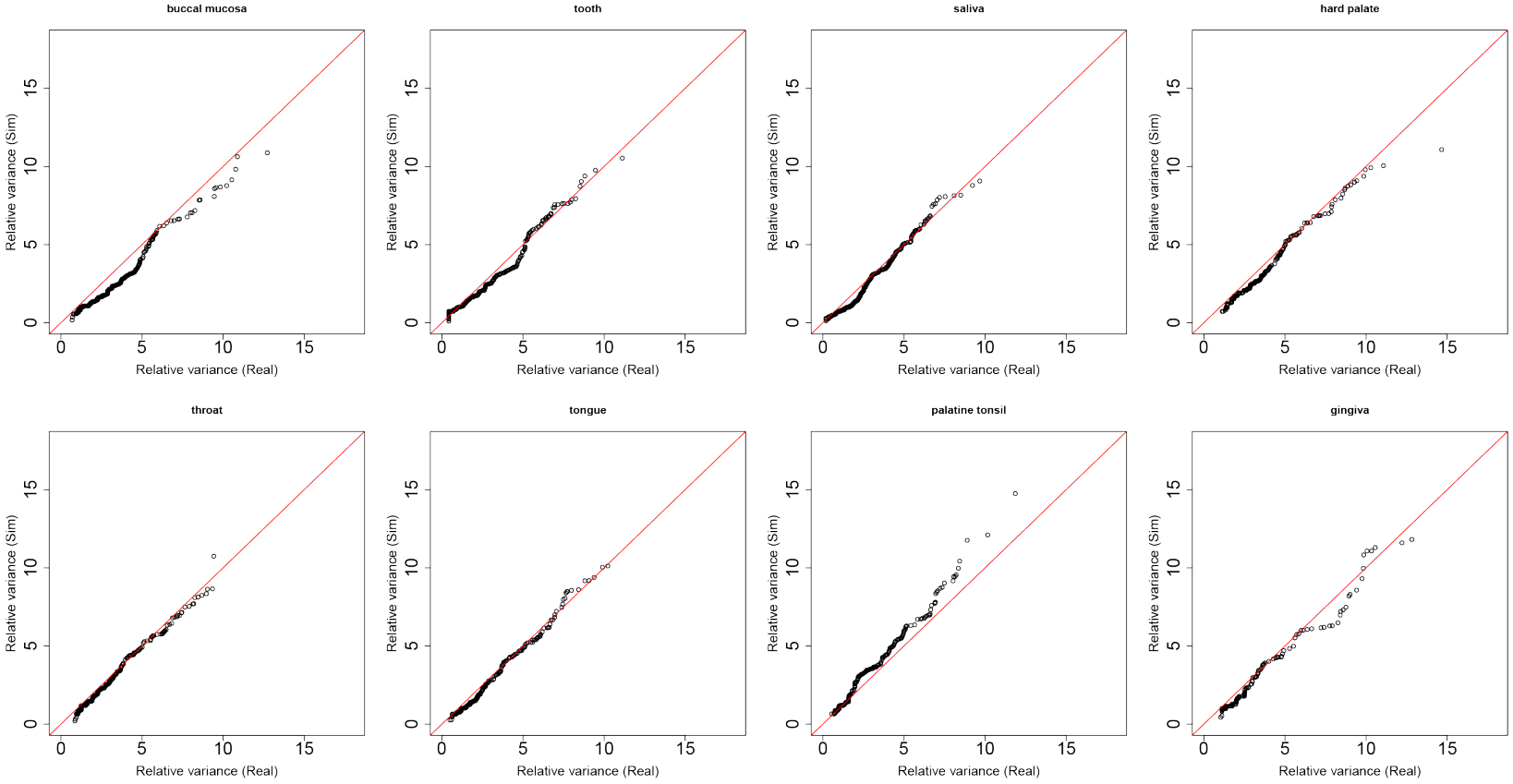


Fig. 6.20 Q−Q plots of of Log2 RV values in real and simulated datasets within each group for HMP dataset, excluding zero mean features.

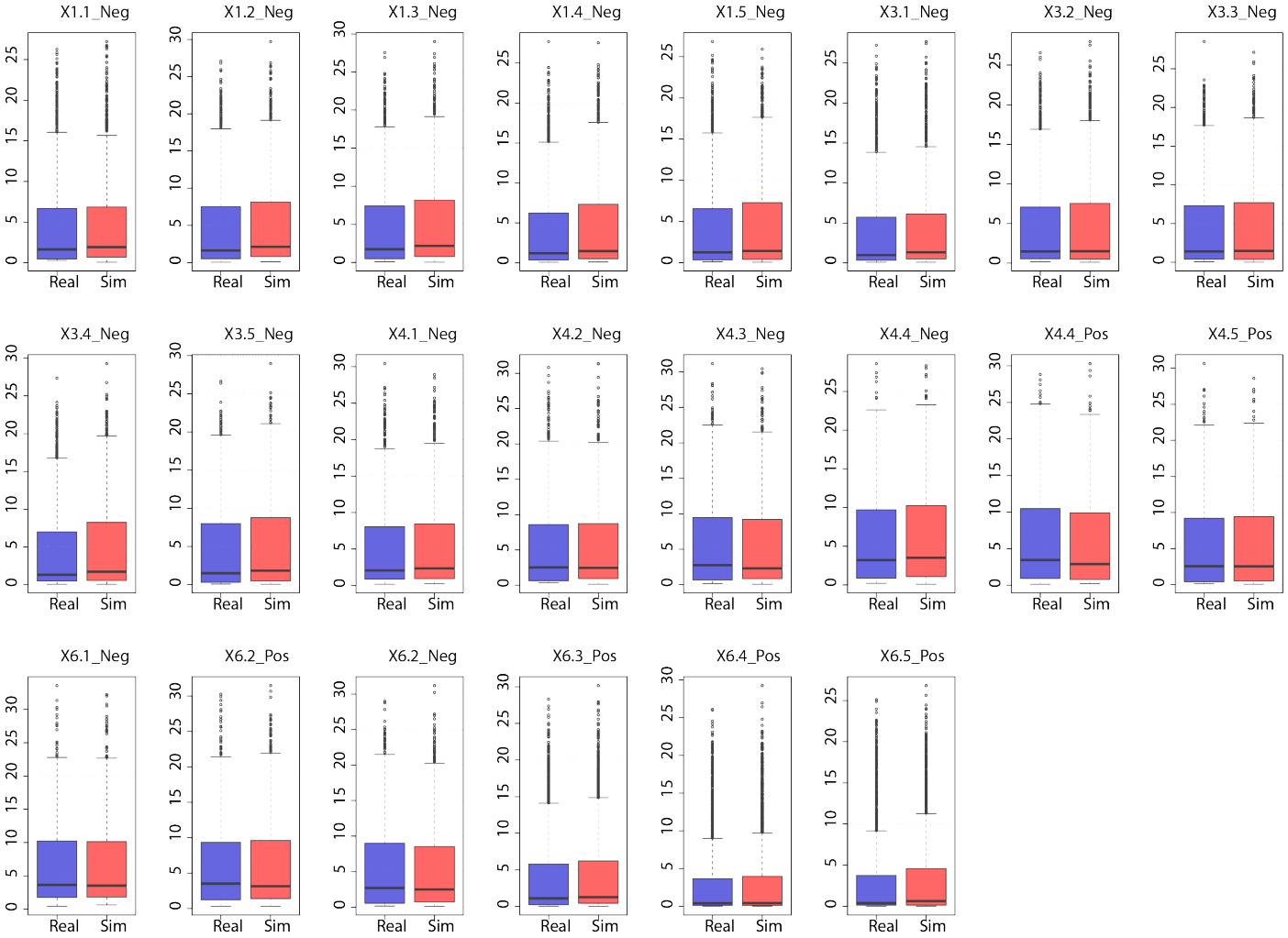


Fig. 6.21 Box plots of of Log2 variance values in real and simulated datasets within each group for animal gut dataset, excluding zero mean features.

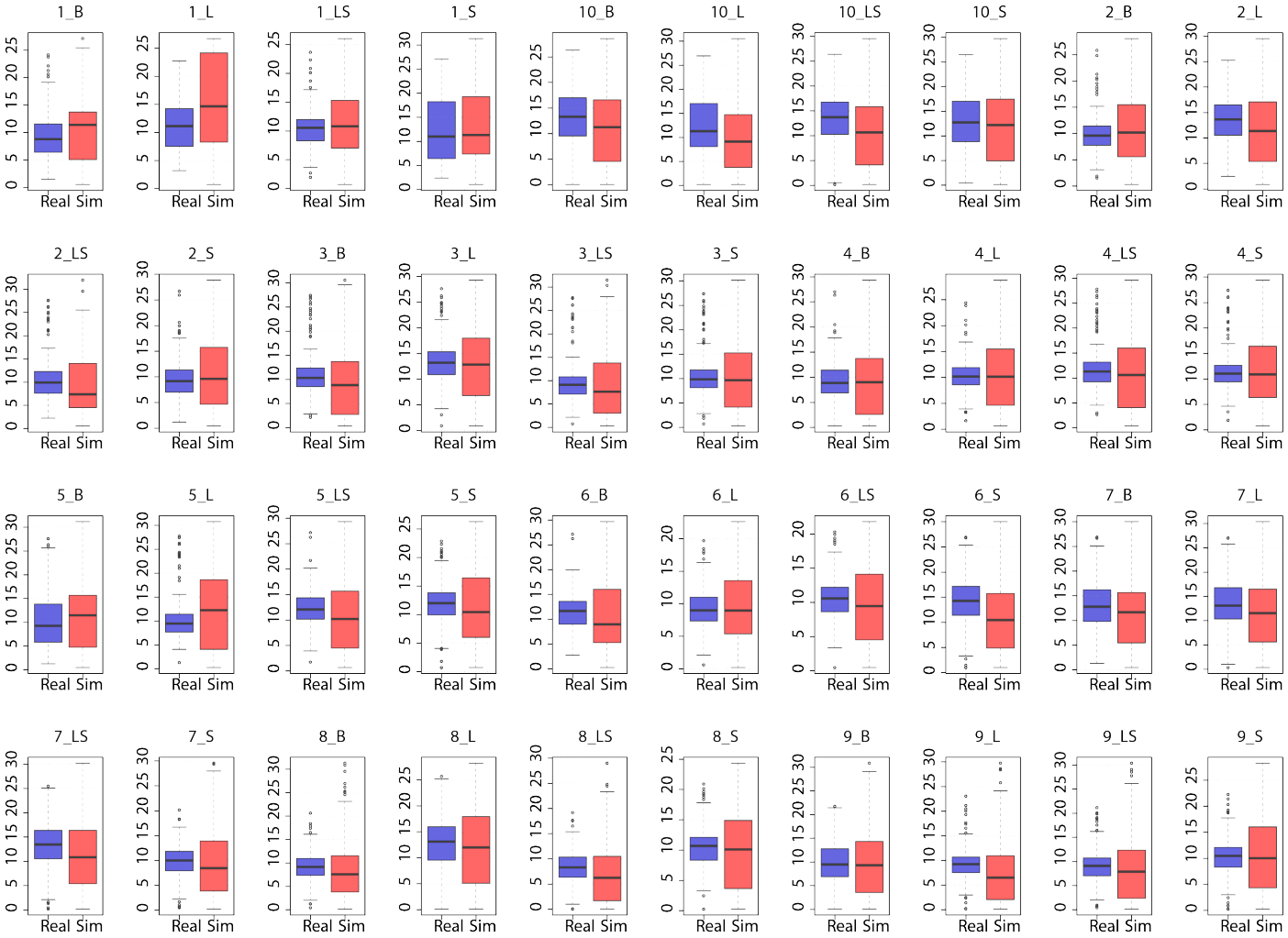


Fig. 6.22 Box plots of of Log2 variance values in real and simulated datasets within each group for raw milk cheese dataset, excluding zero mean features.

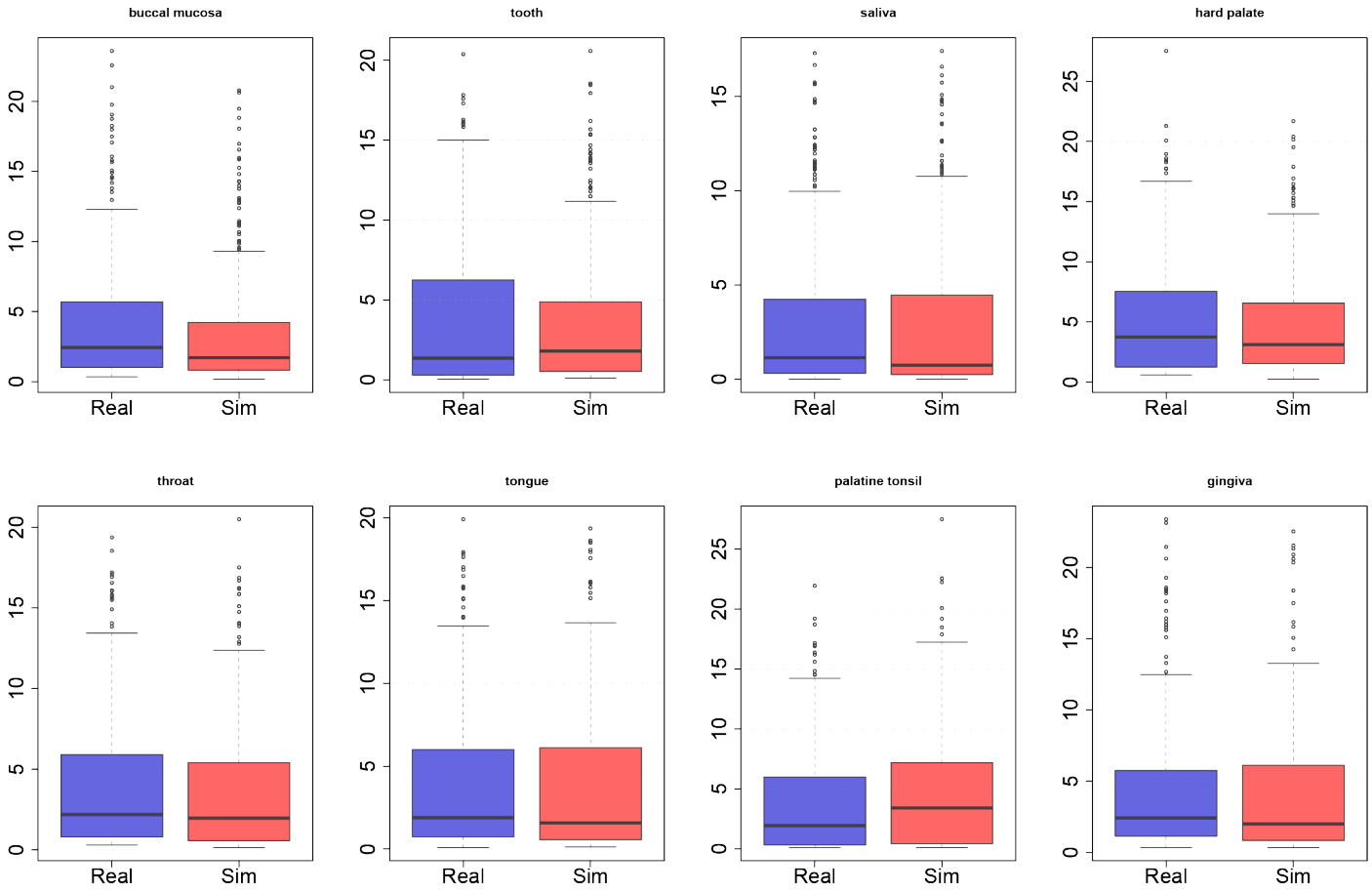


Fig. 6.23 Box plots of of Log2 variance values in real and simulated datasets within each group for HMP dataset, excluding zero mean features.

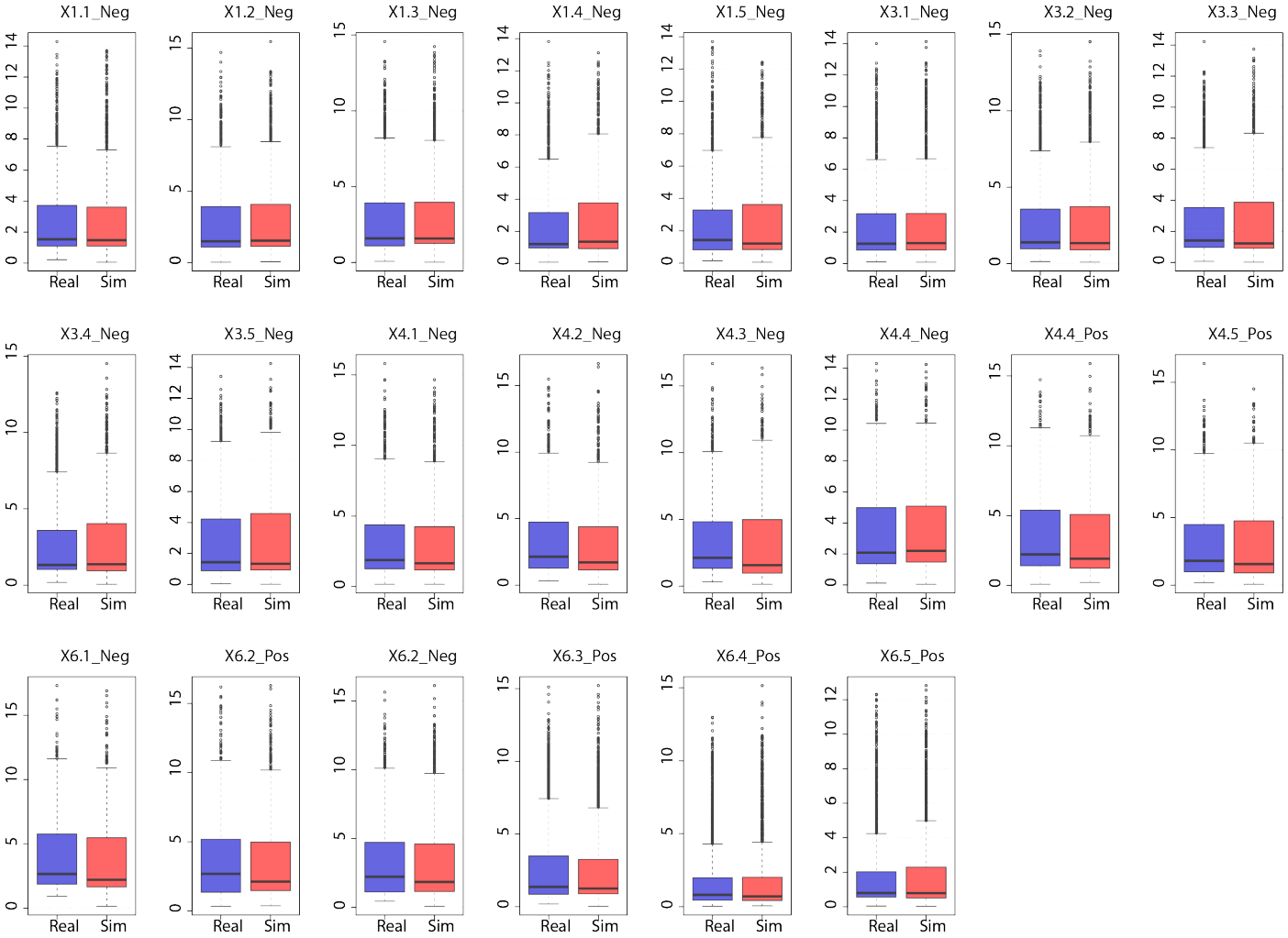


Fig. 6.24 Box plots of of Log2 RV values in real and simulated datasets within each group for animal gut dataset, excluding zero mean features.

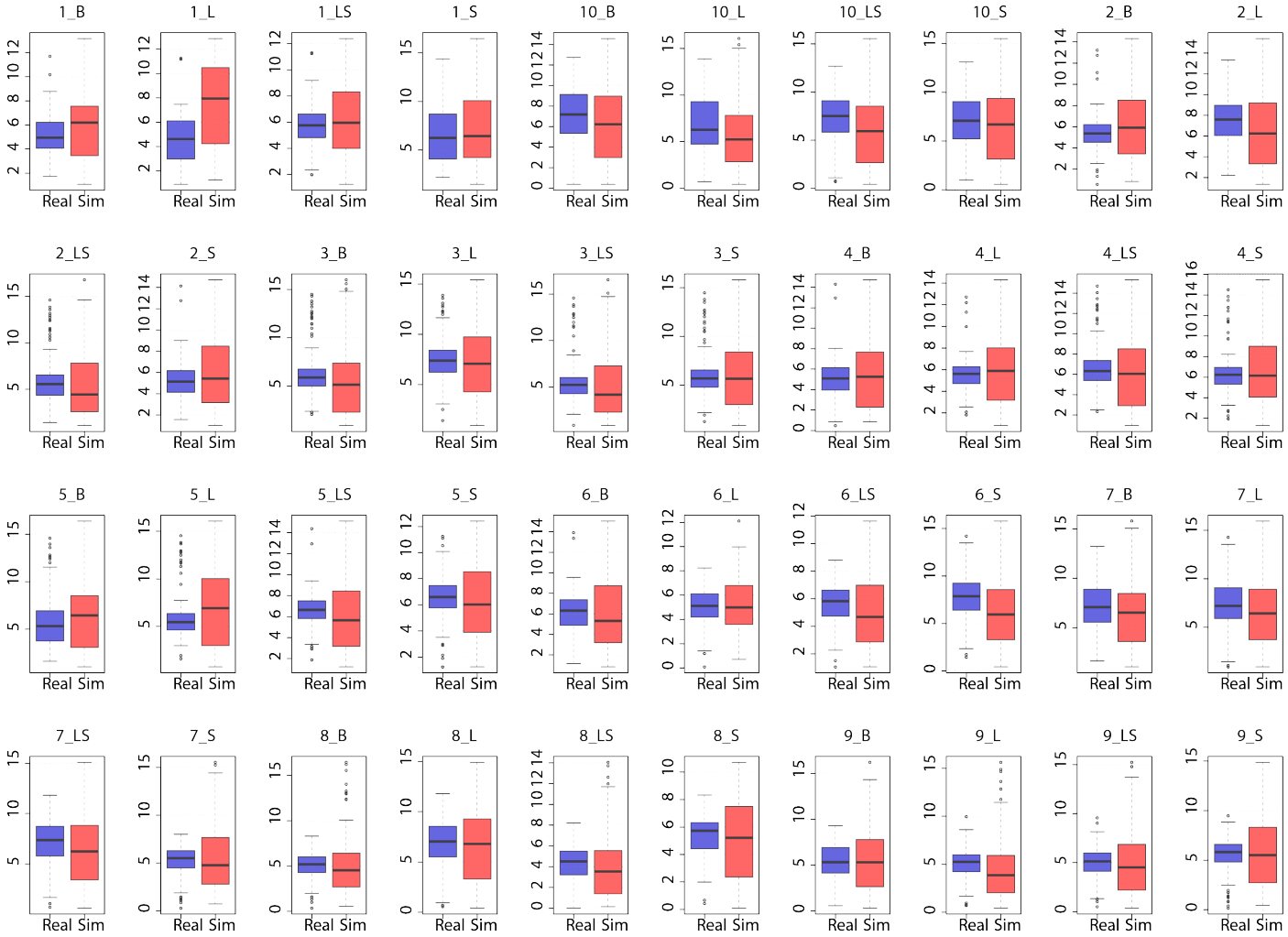


Fig. 6.25 Box plots of of Log2 RV values in real and simulated datasets within each group for raw milk cheese dataset, excluding zero mean features.

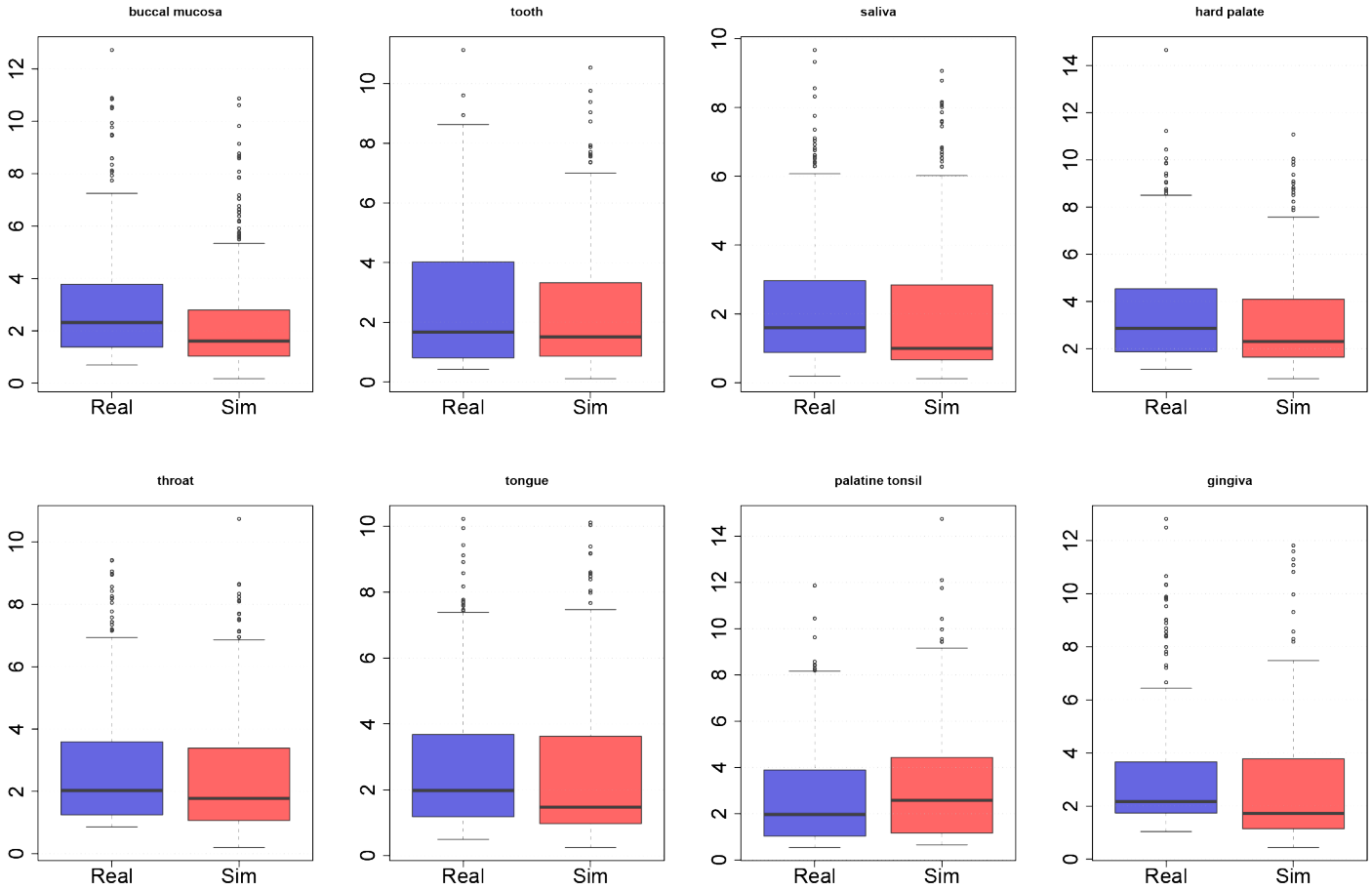


Fig. 6.26 Box plots of of Log2 RV values in real and simulated datasets within each group for HMP dataset, excluding zero mean features.

Table 6.4 Mann-Whitney U tests and effect size results in comparing real and simulated variance distributions within groups for animal gut dataset.

Group	*P* value	Significance	Cohen's *d* magnitude	Bootstrap significance (%)
1	0	Y	Negligible	1.71
2	0	Y	Negligible	2.17
3	0	Y	Negligible	7.82
4	0	Y	Negligible	1.67
5	0.11	N	—	0
6	0	Y	Negligible	1.22
7	0.517	N	—	0.01
8	0.313	N	—	0
9	0	Y	Negligible	1
10	0	Y	Negligible	1.97
11	0.03	Y	Negligible	0.14
12	0.851	N	—	0
13	0.403	N	—	0
14	0.038	Y	Negligible	0.08
15	0.083	N	—	0.03
16	0.257	N	—	0
17	0.241	N	—	0
18	0.401	N	—	0.06
19	0.274	N	—	0.08
20	0	Y	Negligible	0.84
21	0.041	Y	Negligible	0.03
22	0	Y	Negligible	0.6

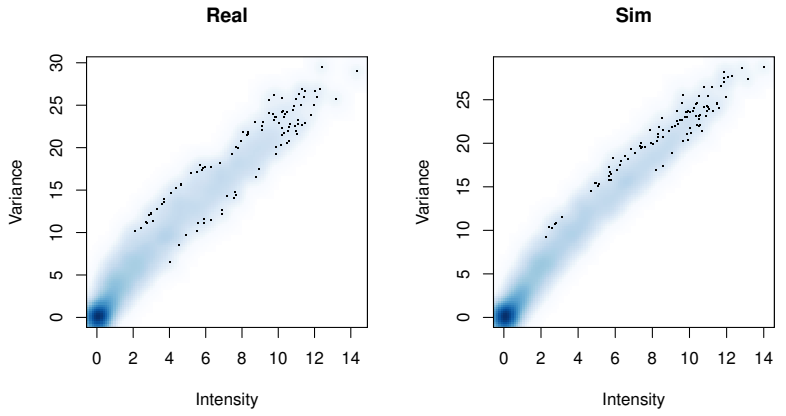
Table 6.5 Mann-Whitney U tests and effect size results in comparing real and simulated variance distribution within groups for raw milk cheese dataset.

Group	*P* value	Significance	Cohen's *d* magnitude	Bootstrap significance (%)
1	0.004	Y	Negligible	0.65
2	0.113	N	– – –	0.04
3	0	Y	Negligible	5.86
4	0.008	Y	Negligible	0.47
5	0.001	Y	Negligible	0.17
6	0	Y	Negligible	2.13
7	0	Y	Negligible	0.92
8	0	Y	Negligible	0.57
9	0	Y	Negligible	2.56
10	0	Y	Negligible	6.36
11	0	Y	Negligible	4.12
12	0	Y	Negligible	1.25
13	0	Y	Negligible	8.87
14	0	Y	Negligible	3.06
15	0	Y	Negligible	7.69
16	0	Y	Negligible	3.95
17	0	Y	Negligible	4.88
18	0.003	Y	Negligible	0.66
19	0.002	Y	Negligible	0.34
20	0	Y	Negligible	3.54
21	0.005	Y	Negligible	0.75
22	0	Y	Negligible	2.17
23	0	Y	Negligible	1.4
24	0	Y	Negligible	3.64
25	0	Y	Negligible	1.8
26	0	Y	Negligible	2.81
27	0	Y	Negligible	2.26
28	0	Y	Negligible	2.07
29	0	Y	Negligible	7.29
30	0	Y	Negligible	3.45
31	0.001	Y	Negligible	0.19
32	0	Y	Negligible	1.98
33	0	Y	Negligible	4.85
34	0	Y	Negligible	0.48
35	0	Y	Negligible	0.68
36	0.002	Y	Negligible	0.44
37	0	Y	Negligible	0.48
38	0	Y	Negligible	4.09
39	0.002	Y	Negligible	0.15
40	0	Y	Negligible	0.98

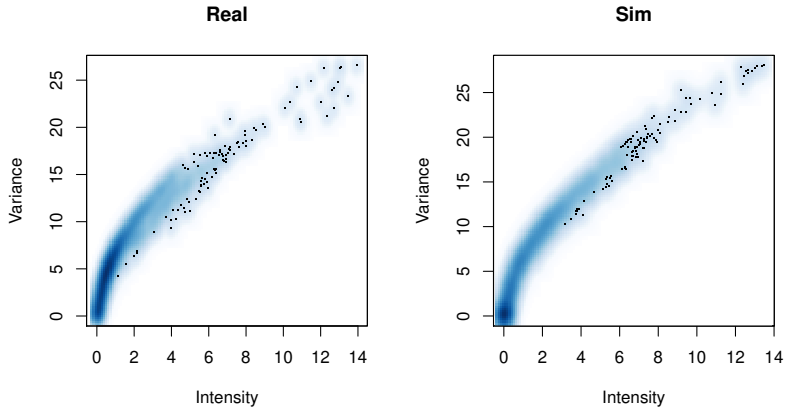
Table 6.6 Mann-Whitney U tests and effect size results in comparing real and simulated variance distribution within groups for HMP dataset.

Group	*P* value	Significance	Cohen's *d* magnitude	Bootstrap significance (%)
1	0	Y	Negligible	0.09
2	0.017	Y	Negligible	0.01
3	0.001	Y	Negligible	0.06
4	0.007	Y	Negligible	0.02
5	0.007	Y	Negligible	0
6	0.105	N	– – –	0
7	0	Y	Negligible	0.67
8	0.08	N	– – –	0.01

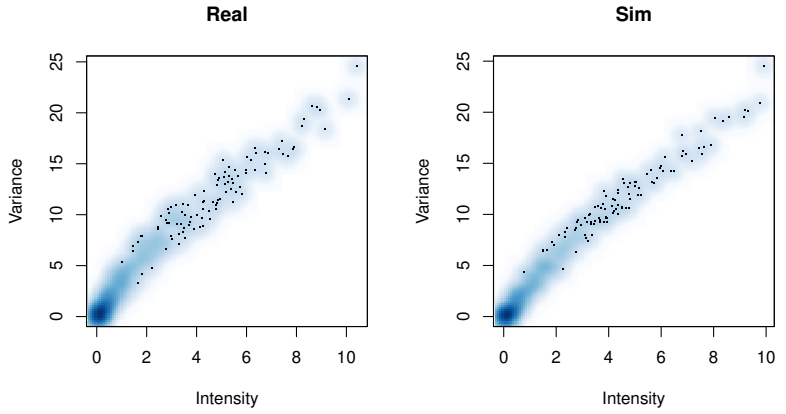
Lastly, the goodness in recreating the intensity-variance relation among counts was investigated for all the datasets. The results (Figure 6.27) showed a very good recreation of real data characteristics, with raw milk cheese dataset being the less precise estimate. Nevertheless, also in this testing situation a good overall shape reconstruction was obtained.



(A)



(B)



(C)

Fig. 6.27 Scatter plot of the relation between Log2 count mean intensity and variability in real and simulated datasets, for animal gut (first row), raw milk cheese (second row) and HMP (third row) data, excluding zero mean features.

6.4 Conclusions on performance results

We tested our 16S rDNA data simulator, metaSPARSim, into three very different conditions. In the first dataset, a good number of biological replicates (five) was available for each group and a high but not extreme sparsity level (78.7%) characterized the data. In this framework, metaSPARSim was able to reconstruct real data properties almost perfectly, in terms of all the three features (sparsity, intensity and variability) considered. Also inter-feature relations, such as intensity-sparsity and intensity-variance relations, were faithfully reproduced.

Regarding food microbiota (raw milk cheese) dataset, only 2 or 3 replicates per group were available and global sparsity level was dramatic (97.1%). In these peculiar conditions, obtaining an estimate of reliable starting parameters is a very challenging task. This reflected in a little worsening of performance in the simulation results, but the simulator still guaranteed a realistic output, maintaining the principal features realistic in median into an acceptable range of variation.

The third dataset (HMP data) put no particular difficulty linked to the number of replicates per group (5/100) or sparsity level (80%), but to their nature. More precisely, the fact that replicates came from different subjects added interindividual variability to the natural biologic one. In fact, groups in this case were not made by experimental biological replicates, i.e. biological resamplings of the same sample type, but by samples coming from different individuals in the same body sites. When groups represent sampling sites known to be highly linked to individual characteristics, e.g. human oral bacterial community, the Gamma distribution may not be the most appropriate modelling to perfectly describe whole variability. However, tests on both little and large population taken from HMP data revealed the robustness of the simulator to this situation, proving its usability also in simulating this kind of data. In fact, simulated sparsity, intensity and variability correctly mimicked the real ones, thus assuring a valid synthetic dataset for developers who want to use it to test their tool performance on a human-microbiome-like dataset.

Chapter 7

Benchmark of tools for 16S rRNA gene sequencing data pre-processing

In Chapter 3 we saw in detail some of the most used approaches and tools for sequencing count data pre-processing. As explained before, those tools are taken from different contexts, such as RNA-Seq or scRNA-Seq, but most of them have been along the years transferred from their native environment and widely used also for amplicon sequencing data analysis. In fact, the lack of specific tools for 16S rDNA sequencing data pre-processing and the sharing of some distinctive traits with other sequencing data types led to a direct use of such tools in this different context. However, it should be considered that the choice of the most adequate pre-processing pipeline to perform an analysis is strictly linked to the characteristics and the nature of analyzed data. Consequently, before applying a method that was created for a different context from 16S data analysis, an assessment of its performance in a controlled (simulated) situation should be performed. The benchmark of tools introduced in Chapter 3 and the identification of an optimal pipeline formed buy a juxtaposition of a subset of them has been examined in this thesis to address this issue and to provide useful indications to 16S rDNA data analysts on advantages and drawbacks of the most used and recent pre-processing tools available in literature.

7.1 Pipelines and ground truth

The previously introduced and disclosed tools were combined to form 48 different pre-processing pipelines. The first group of them represent the most frequently adopted approach to analyze 16S data that consists solely in the normalization step. The six approaches and tools that compose this first group are the ones presented in Chapter 3 and will be referred

to in the following as *TSS*, *CSS*, *edgeR*, *Deseq2*, *scran* and *GMPR*. The second portion of pipelines are uniquely formed by the six imputation methods considered in this thesis and will be referred to as *DrImpute*, *scImpute*, *LLSimpute*, *LowRank*, *zCompositions_SQ* and *zCompositions_CZM*. The last group of pipelines is composed by all the combinations of the previous normalization and zero-imputation approaches. Even if some imputation tool explicitly asked for raw or normalized data, the performance of each tool was considered when used both singularly and in combination with a normalization step, to investigate if some normalization technique could affect positively even tools initially thought to deal with raw data. This group of pipelines were labelled with the juxtaposition of the normalization and the zero-imputation approach names, separated by an underscore. For example, in the following we will refer to the dataset normalized with *TSS* and then processed with *DrImpute* as "TSS_DrImpute".

All the pipelines results were compared with a dataset that was considered to be the ground truth for real (unobserved) abundances in the samples. This data represent pre-sequencing abundance values. i.e. "true" abundances in samples prior to sequencing. The pipelines were then tested for goodness in retrieving information and characteristics of data prior to the possible loss of information due to under-sampling.

7.2 Datasets

Three simulated datasets were selected for benchmarking (Table 7.1). The first derives from animal gut microbiota dataset and was obtained taking 14 out of the 22 available in the rial data, without less of generality and for clearer graphical results. The second one was obtained starting from parameters estimated from the food microbiota experiment, also reducing groups to a more treatable subset (12 out of 40). Finally, a third dataset was generated with parameters obtained from HMP data, maintaining the 8 initially selected groups. In the following all the related details are reported.

Dataset 1: this dataset was simulated taking as input the parameters estimated from the chicken gut microbiota dataset. The intensity vector for each group was calculated as the mean of the five available replicates, while the dispersion was estimated using *edgeR* dispersion estimation from rows with three or more non-zero values. This filter was introduced here and in the other datasets dispersion estimate because *edgeR* was found to possibly introduce biases in presence of a too high sparsity level. To mitigate this effect, a filter on OTUs for each group was applied for robust dispersion estimation. The simulated dataset was characterized by a 13.2% of total zeros being introduced by the sequencing process, that caused a pre-sequencing sparsity of 63.03% to rise to a

raw data sparsity of 72.56%. The drop in sparsity level compared to real data is due to the exclusion from the simulation of about one third of the group. As a consequence, OTU rows that had some non zero values in the original data but had only null values in the restricted one were deleted, with a little drop in the sparsity descriptive statistic. In addition, twice the number of original replicates were generated, with the level of sparsity being lowered consequently. The specifications of data structure are reported in the following.

Dataset 2: the second dataset was simulated starting from HMP data. Also in this case, the intensity vector for each group was calculated as the mean of the available replicates, while the dispersion was estimated using *edgeR* dispersion estimation from rows with 20% or more non-zero values. Also in this case, the number of replicates was increased and this caused the sparsity to drop from about 81.36% to 67.91%.The obtained synthetic dataset was then characterized by a 16.6% of total zeros being introduced by the sequencing process, that changed a pre-sequencing dataset with a sparsity of 56.61% into a raw data matrix with a sparsity of 67.91%. The specifications of data structure are reported in the following.

Dataset 3: the last dataset was obtained from raw milk cheese microbiome data. Again, the intensity vector was calculated as biological replicates mean and the dispersion estimate was obtained by the use of *edgeR* estimation with threshold for OTU retention of 30% or more non-null values. The simulated dataset was characterized by a total sparsity of 94.34%, with only 3.26% of which attributable to sequencing process. This little percentage is well-explained if we consider the experiment from which estimated parameter for simulation came from. In fact, the food microbiome sequencing was performed with a very high sequencing depth per sample. This datum, jointly with the modest (with respect to, for example, gut microbiome niches) diversity of the microbial population that typically populate food matrices, explain why the simulated data showed us a very uneven division of sparsity between the one due to sequencing and the remaining, the latter being the vast majority of the total. This dataset was thought as a good and challenging testing situation for pipelines that included the zero-imputation step. In fact, this dataset mimicked an ideal situation in which almost all the information was caught (little sequencing loss of information) and then permitted to assess possible biases introduced by hypothetically unnecessary imputation usage.

Table 7.1 Simulated dataset characteristics

	Dataset 1	Dataset 2	Dataset 3
Groups	14	8	12
Samples	140	80	120
Replicates	10	10	10
Features	3326	758	1140
Sequencing depth (range)	16347-995050	2763-97612	30165-293285

7.3 Evaluation criteria

In this section, the methods and metrics through which pipelines performance was evaluated are introduced and explained. The adopted benchmarking framework, represented in Figure 7.1, involved ground truth data jointly with raw and pre-processed data. Starting from the ground truth parameters, raw count tables were produced by the use of metaSPARSim. Then, raw matrices were pre-processed with all the 48 pipelines considered in this thesis, that include 6 normalization-only and 6 zero-imputation-only pipelines and 36 pipelines combining all the normalization and zero-imputation methods. Finally, ground truth, raw and pre-processed data were evaluated according to a set of criteria that are explained in the following.

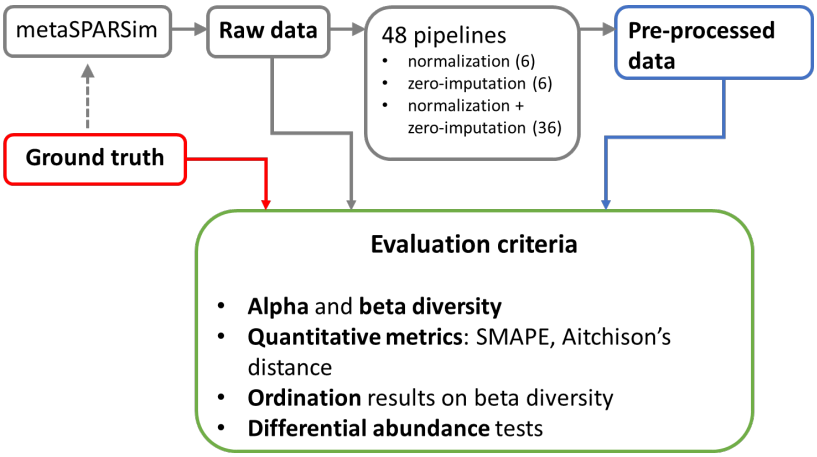


Fig. 7.1 Benchmarking framework.

7.3.1 Total sparsity recovery

As a first metric, the pipelines were evaluated for their performance on recreating original data sparsity. Each pipeline results were compared to the ground truth, i.e. data before

sequencing. Pre-processed data were then ranked for distance to real sparsity level and labelled according to their under("U")- or over("O")-estimation behaviour.

7.3.2 Proportional abundances reconstruction

To assess the ability in recovering "true" (ground truth) proportional abundances, two different metrics were considered: SMAPE and Aitchison's distance.

The first accuracy measure, called *symmetric mean absolute percentage error (SMAPE)* is a measure based on percentage (or relative) errors. More precisely, it quantifies the error between two vectors x and y as:

$$SMAPE(x,y) = \frac{100}{n}\sum_{i=1}^{n}\frac{|x_i - y_i|}{|x_i| + |y_i|}. \tag{7.1}$$

For each sample, the ground truth and the pre-processed data vector were compared using SMAPE measure. As an aggregating function, the median of SMAPE values across samples was calculated for each dataset, thus giving an overall quantification of each pipeline accuracy. This relative error measure was chosen as an alternative to the classical relative error because of heavy data sparsity. In fact, when the reference value in the relative error formula is zero, this measure becomes undefined. Additionally, to overcome the undefined form obtained in SMAPE when both ground truth and examined value were both null, a SMAPE value of zero was imposed. The second measure of goodness between true and pre-processed proportional abundances, Aitchison's distance [88], accounts for the compositional nature of data, thus becoming more appropriate respect to the classically adopted Euclidean distance-based methods to measure differences in count data. In fact, as seen in Chapter 3), Euclidean distance violates several simple principles of compositional data analysis, such as scale invariance, perturbation invariance, and subcompositional dominance. We recall here the explicit formula of this distance for two generic vector x and y is :

$$d_A(x,y) = \sqrt{\frac{1}{D}\sum_{i=1}^{D-1}\sum_{j=i+1}^{D}\left(ln\frac{x_i}{x_j} - ln\frac{y_i}{y_j}\right)}. \tag{7.2}$$

As for SMAPE metric, an aggregated result for each dataset was produced by taking the median of Aitchison's across samples.

7.3.3 Impact on bacterial diversity

One of the most important aspects that have to be taken into account when performing a microbiome analysis is the investigation of population diversity. Diversity indices created to this purpose are mathematical measures used to investigate the dynamics or to detect differences in the composition of the ecological niche (sample) between different conditions. However, as well described by Finotello et al. [110], diversity is not a determined physical quantity for which a consensus definition and unit of measure have been established, and several diversity indices are currently available. These indices are generally divided into two categories: alpha and beta diversity indices. These terms were originally introduced by the ecologist Whittaker [111] [112] in 1960 to describe the biodiversity of an ecosystem. According to Whittaker, total species diversity in a landscape (called *gamma* diversity) is formed by two contributions: the species diversity in sites belonging to a niche (*alpha* diversity) and the differentiation among those sites (*beta* diversity). Anyway, the mathematical formulation of alpha and beta diversity has not a unique form and also their formal definition has been revised several times and a consensus on that has not been reached yet. Despite some works have attempted to mathematically distinguish total diversity into alpha and beta diversity [113][114], the standard practice is to use separately alpha and beta diversity indices to estimate a sort of intra- and inter-sample diversity, respectively. In particular, the first is used to describe the compositional complexity of a single sample, while beta diversity is commonly intended as a measure of taxonomical differences between samples. As a qualitative argumentation, one can state that a sample has high alpha diversity when it contains a high number of equally abundant species, and low diversity otherwise, whereas when comparing two samples, beta diversity has high values if the compared populations share few species and a low one if most of their species are in common. Due to the lack of a univocal and established definition of diversity and the multifaceted nature of this concept, several mathematical indices where introduced in time which have been originally applied to ecology and, later, broadly used in the analysis of 16S sequencing data. These indices have different purposes, units of measures and mathematical formulations and this implies that the use of more than one index is recommended to look at different population characteristics when performing a bacterial population analysis. In this thesis, five alpha and two beta diversity measures were considered to assess each pre-processing pipeline effects on microbial community composition, with the aim of identifying the ones that would preserve the most the real (usually unknown) structure of pre-sequencing data. The details of the selected indices are reported in the following, as well as their mathematical formulations.

7.3.3.1 Alpha diversity

As introduced before, we can qualitatively define alpha diversity (also called *local diversity*) as a quantification of the compositional complexity of an ecosystem, which increases with the number of present species and the evenness of their proportional abundances. All different available indices of alpha diversity are classifiable into three main categories: *richness* indices, which estimate the number of different species in a sample; *evenness* indices, which consider the species relative abundances, without considering their total number; and *diversity* indices, which account for both the species relative abundances and their total number.

Richness indices. Observed richness index (denoted as S_{obs}) provides a direct measure of alpha diversity by simply counting the number of different species present in a sample. As said before, during site sampling and sequencing some rare species can be lost and species richness might then be underestimated. To address this issue, richness estimators that correct observed richness for the number of hypothetically lost species were created. These indices estimate unseen species considering the distribution of the rarest ones, i.e. singletons and doubletons, that are species with exactly one and two counts, respectively. These are, for example, Chao1 [115] and the first- and second-order Jackknife indices [116]. This kind of richness estimators are not of interest and will not consequently be considered in this thesis because only the true number of present species has a meaning when measuring lost information recovery performed by each pre-processed pipeline. In fact, normalized and imputed data are transformed data that loose the integrity characteristic of count data, making nonsensical to look for singletons and doubletons existence. In addition, the species loss issue is already the main focus of these pre-processing pipelines and using corrected richness indices would mean to double addressing this issue.

Evenness indices. Evenness indices measure how evenly the relative abundances are distributed across the different species. Thus, they are both a valuable indicator of biodiversity and an effective way to determine the stability and resilience of an ecosystem. A commonly used evenness measure is the evenness factor ${}^{q}EF$, defined as

$$ {}^{q}EF = \frac{\left(\sum_{i=1}^{S_{obs}} p_i^q\right)^{\frac{1}{1-q}}}{S_{obs}}, \tag{7.3} $$

where the numerator identifies the so-called Hill numbers of order q (designed as ${}^{q}D$) [117] and p_i are species relative abundances. It is noteworthy that from this definition it is impossible for two communities with minimal evenness (i.e. ${}^{q}D_1 = {}^{q}D_2 = 1$), but different S_{obs} (e.g. $S_{obs1} = 10$ and $S_{obs2} = 1000$), to obtain the same EF. In particular,

the community with more species will always have the lowest EF. For this reason, indices of relative evenness (RLE), which scale to the range of values that are possible for a given richness S_{obs}, have been developed. In particular, in this thesis the Pielou index was chosen to measure sample evenness. It considers the logarithm of Hills numbers of the first order and divides it by the logarithm of total observed species, i.e.:

$$Pielou = \frac{ln(^1D)}{ln(S_{obs})}, \tag{7.4}$$

Diversity indices. As introduced above, complex alpha diversity indices can be computed taking into account the relative abundance p_i of each species and considering the yet introduced Hill numbers of order q (^qD). By equivalently re-writing ^qD as

$$^qD = \frac{1}{\left(\sum_{i=1}^{S_{obs}} p_i^{q-1} p_i\right)^{q-1}}, \tag{7.5}$$

it can be noticed that ^qD gives a weighted mean of species abundances and that the order q defines the type of weighted mean computed at the denominator. For example, for $q = 0$, the harmonic mean is computed; for $q = 1$, the geometric mean; and for $q = 2$, the arithmetic mean. Furthermore, by increasing the value of q, the weight given to the most abundant species with respect to the rarest ones increases. That is, for $q = 0$, species weights cancel out species abundances and the index is equivalent to S_{obs}, as the sum is computed only on present species with $p_i > 0$. The most widely used diversity indices [118] are Shannon entropy and inverse Simpson index (or Simpson concentration), that can be easily obtained starting from Hill numbers. In fact, Shannon entropy (H) is equal to the logarithm of 1_D, i.e.

$$H = -\sum_{i=1}^{S_{obs}} p_i ln(p_i), \tag{7.6}$$

while inverse Simpson index (IS) is equal to ^2D, i.e.

$$IS = \frac{1}{\sum_{i=1}^{S_{obs}} p_i^2}. \tag{7.7}$$

These indices differ in their theoretical foundation and interpretation. In fact, H has its foundations in information theory and represents the uncertainty about the identity of an unknown individual. In a highly diverse and evenly distributed sample, an unknown individual could belong to any species, leading to a high uncertainty in predictions of

its identity. In a less diverse niche, dominated by one or a few species, it is easier to predict the identity of unknown individuals and there is less uncertainty in the system. On the other hand, *IS* represents the inverse of the probability that two randomly chosen individuals belong to the same species. They thus give some different information and different weight to most abundant species, the second giving more importance to most abundant species with respect to the rarest ones.

In addition to these two popular indices, the recently proposed Tail statistic [119] was considered in this thesis to improve sensitivity in the presence of rare species. In fact, this is of particular interest when benchmarking pipelines intended for rare species information loss. Its mathematical formula is the following:

$$Tail = \sqrt{\sum_{i=1}^{S_{obs}} p_i (i-1)^2}, \tag{7.8}$$

with $p_1 \geq p_2 \geq \cdots \geq p_{S_{obs}}$.

7.3.3.2 Beta diversity

As discussed above, beta diversity is commonly used in 16S sequencing studies to highlight species differences between pairs of samples. A key distinction is between beta-diversity metrics that use presence-absence data and metrics that use species abundances. Even if abundance data are clearly more information-rich than binary data, species abundances are usually not taken into account and only presence-absence data are used to identify which species are shared by samples and which are not. Since Whittaker's original suggestion [111][112] that beta diversity should be measured as the proportion by which the species richness of a region exceeds the average richness of a single locality within that region, numerous measures have been proposed that constitute variations on this them, some of which are inherently closely correlated, while others give wildly different patterns of results for the same data sets. In general, however, the reasons for using, or preferring, any particular measure rather than another remain unclear. For this reason, in this thesis beta diversity on presence−absence data was calculated using Whittaker's original measure, that currently remains the most frequently employed beta diversity measure.

In particular, if the list of present species in two samples is usually expressed in terms of components a, b and c, where a is the number of species shared by both samples, and b and c are the number of species present respectively only in the first or in the second sample, the total number N of species accounted by a pair of samples can be written as $N = a + b + c$.

Now, Whittaker beta diversity can be expressed as

$$\beta_w = \frac{b+c}{2a+b+c}. \tag{7.9}$$

Considering abundance-based measures, the most widely used (so that it is the default abundance-based beta index in the well-known *vegan* [120] *R* package) is the Bray-Curtis index. If we consider two generic vectors of abundances, x and y, it is defined as

$$d_{Bray-Curtis}(x,y) = \frac{\sum_i |x_i - y_i|}{\sum_i (x_i + y_i)}. \tag{7.10}$$

This measure was chosen for abundance data beta diversity calculation and used in combination with Whittaker's presence−absence measure to compare the ground truth and pre-processed data structure.

7.3.4 Impact on differential abundance analysis results

Differential abundance analysis is a fundamental step in each microbiome study. It consists in identifying possible features that differ in their presence between two sample categories, e.g. the body site from which a sample was collected or the geographical area from which a soil sample was taken. Differentially abundant species are identified by statistical tests that check for their significant difference in abundance between groups of samples. This step is well-implemented in some available *R* packages for metagenomic data analysis; however, in this thesis we choose to perform it using a non-parametric Mann-Whitney test. This decision was based on the fact that each differential analysis tool has its own assumptions and underlying model, so we preferred a neutral evaluation method that was independent on the choice (and goodness) of data modellization.

In particular, for the ground truth, the raw and each pre-processed dataset and for each feature, a paired Mann-Whitney U test was performed between data coming from different conditions. Differentially abundant features were identified for each conditions comparison by selecting the resulting P values that fall under the significance level of 0.05 after Benjamini-Hochberg FDR correction for multiple testing [121]. Then, as a concordance measure on results between ground truth and all other datasets for each comparison, Jaccard index (also known as *Intersection over Union*) [122] was used:

$$I_j = \frac{GT_{ij} \cap D_{ij}}{GT_{ij} \cup D_{ij}}, \tag{7.11}$$

where GT_{ij} is the set of differentially abundant features for conditions i and j of ground truth data and D_{ij} is the correspondent set of features identified as differentially abundant in the generic raw or pre-processed dataset D.

From all the group-group comparisons, a set of Jaccard values for concordance with ground truth results was obtained for each dataset. On these values, a one-sided Mann-Whitney test was performed between raw data concordance values and each of the pre-processed data to test for improvement in concordance results between no (raw data) and different pre-processing pipelines. In addition, as suggested by Sullivan and Feinn [123], an effect size calculation was coupled with the above statistical test to measure the magnitude of possible significant differences between distributions. Effect size results were then compared using Table 5.3 introduced in Chapter 5.

Chapter 8

Results of the benchmark of pre-processing pipelines on 16S count data

In this chapter, the results of tests on the pipelines formed by all the combinations of tools presented in Chapter 3 are exposed and discussed. In all graphics and tables, the simulated raw count matrix will be referred to as "raw", whereas we will refer to data obtained as output from each pipeline using the juxtaposition of the normalization tool and the zero-imputation one, separated by an underscore. For example, "edgeR_DrImpute" will be the label for the dataset normalized with *edgeR* and then processed with *DrImpute* for zero-imputation. Where no normalization was performed previously to imputation, the label will contain the "None" prefix, while only the name of the normalization tool is used to identify pre-processing without imputation step. The ground truth data obtained from metaSPARSim for comparison are referred to from here on as "true" or "real" values.

Results will be shown separately for each evaluation metric introduced in Chapter 7 to better compare pre-processing pipelines performance when applied to different frameworks (Dataset 1, 2 and 3; see Chapter 7). In the following, we will refer to each framework as "Simulated Dataset" or "Simulation" plus its identifying number, while we will simply use "dataset" as a synonym of the count matrices on which each metric was calculated, i.e. raw counts and matrices obtained from each pipeline application.

As all analyses showed that the greatest contribution to results was introduced by the imputation step rather then by normalization, for a more immediate comparison and for brevity the pipelines performance are here shown in an aggregated way. More precisely, all the results will be shown aggregating outputs according to the used imputation method, reporting for each the mean and standard deviation of results of the pipelines obtained with

the combination with the 7 normalization procedures (no normalization plus the six selected normalization tools). In this aggregation, normalization-only pipelines are referred to as "None", while other pipelines are identified by the imputation method they contain. Detailed results are also reported in Appendix B, where the reader can find specific results obtained for each single pipeline.

8.1 Total sparsity

The first metric used to compare the ground truth with pipelines result was the ability of each approach to recreate data sparsity. In this context, *LLSimpute*, *LowRank* and *zCompositions* with both priors tended to heavily underestimate data sparsity in all the simulated datasets, recovering the majority or also the totality of zero counts (Tables 8.1, 8.2 and 8.3) in combination with all normalization approaches. In Simulation 1, for example, the true sparsity level was 63.03% but *LLSimpute* and *LowRank* pre-processed datasets showed a percentage of zeros of 19.56-27% and 0.15-1.74%, respectively, depending on the normalization method used before zero-imputation (Appendix B). Moreover, *zCompositions*-treated datasets were completely imputed, with no null value left after pre-processing. Similar results were found for Simulated Datasets 2 and 3, where sparsities linked to *LLSimpute* and *LowRank* methods varied their range slightly and *zCompositions* always produced matrices with all non-zero values (Tables 8.2 and 8.3). On the contrary, *scImpute* and *DrImpute*, in combination with each normalization tools, recreated true sparsity very well, slightly overestimating or underestimating the true zero counts, depending on the Simulation. In fact, in Simulation 1 *DrImpute* pipelines were overall the best-performing pre-processing approaches, with an estimated sparsity range of 62.65-62.74%, followed by *scImpute* pipelines that slightly overestimated the total sparsity (69.50-69.51%), while in Simulated Datesets 2 and 3 *scImpute* overperformed *DrImpute* pipelines (Appendix B). Lastly, normalization-only pre-processing pipelines inherently did not act on sparsity, thus returning sparsity equal to the raw matrix level. The above results suggest *scImpute* and *DrImpute* may be valuable tools for recovering zero values, having both suitable information recovery in terms of number of imputed values. *scImpute* turned out to be more stable in defining "false" zero values (sequencing zeros) and accordingly substituting only those counts in all the Simulations, leaving other null counts their original (zero) value. On the contrary, pipelines including *LLSimpute*, *LowRank* and *zCompositions* imputations always overestimated the features in need of imputation, with the latter always changing all zero values. Looking only at this goodness measure, normalization seemed to have little or no effect on the efficiency of the total pipeline, the results of each pipeline including zero-imputation step varying very weakly when choosing

Table 8.1 Simulated Dataset 1, count matrix sparsity. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Real (ground truth) and raw data sparsity are reported on the top-left of the Table.

True: 63.03%		
Raw: 72.56%		
Imputation	**Mean**	**SD**
None	72.56%	3.34%
scImpute	69.50%	0.00%
DrImpute	62.66%	0.03%
LLSimpute	21.32%	9.11%
LowRank	0.83%	0.69%
zCompositions_SQ	0 %	0 %
zCompositions_CZM	0 %	0 %

Table 8.2 Simulated Dataset 2, count matrix sparsity. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Real (ground truth) and raw data sparsity are reported on the top-left of the Table.

True: 56.61%		
Raw: 67.91%		
Imputation	**Mean**	**SD**
scImpute	55.84%	0.00%
None	67.91%	0.00%
DrImpute	42.32%	0.00%
LLSimpute	23.31%	0.86%
LowRank	2.09%	1.82%
zCompositions_SQ	0 %	0 %
zCompositions_CZM	0 %	0 %

a normalization method in place of another one or using no normalization. The distances between sparsity in the ground truth and in the different pipelines are reported in Figure 8.1, where the performance obtained for all the Simulated Datasets is summarized.

Table 8.3 Simulated Dataset 3, count matrix sparsity. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Real (ground truth) and raw data sparsity are reported on the top-left of the Table.

True: 91.26%		
Raw: 94.34%		
Imputation	**Mean**	**SD**
None	94.34%	3.54%
scImpute	87.07%	1.97%
DrImpute	79.89%	1.34%
LLSimpute	23.01%	1.15%
LowRank	3.85%	3.46%
zCompositions_SQ	0 %	0 %
zCompositions_CZM	0 %	0 %

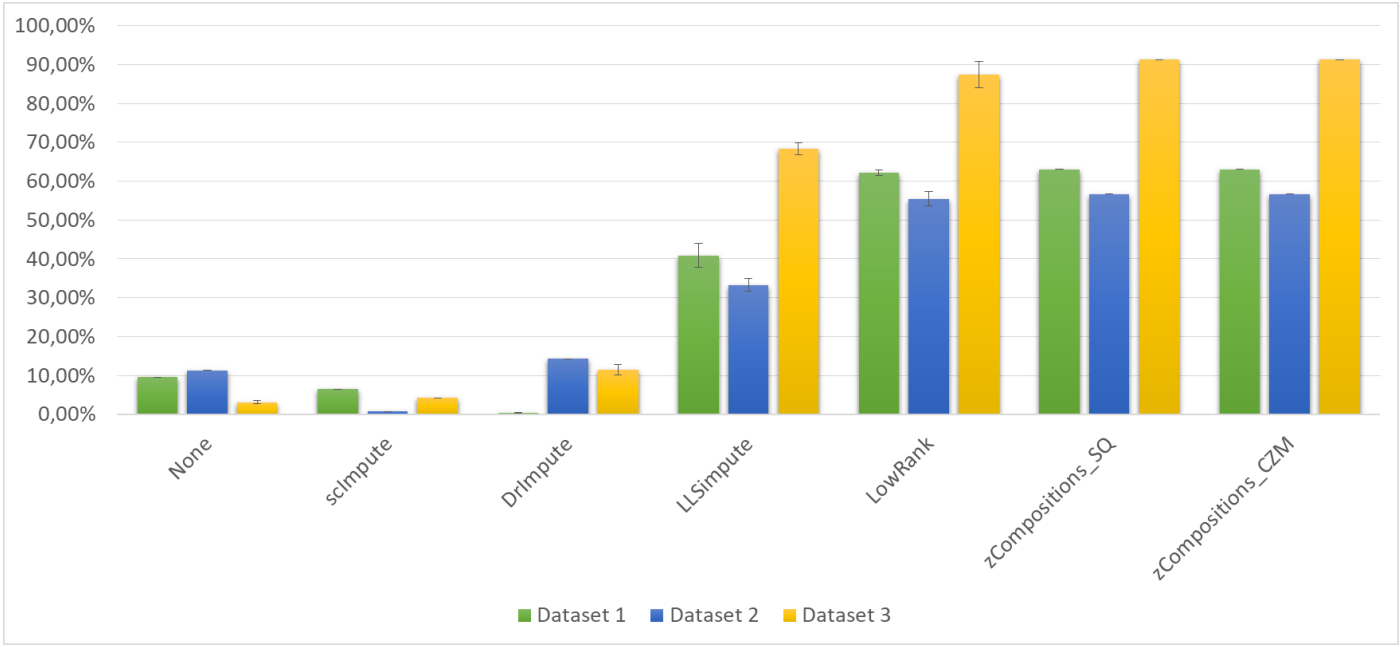


Fig. 8.1 Distances between sparsity in the ground truth and in the different pipelines for the three Simulated Datasets. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

8.2 Relative abundance profile

As introduced in Chapter 7, to evaluate the ability of different pipelines in recovering data information SMAPE and Aitchison's distance were calculated on sample proportional abundances. This permitted to identify which are the pipelines that better respect internal count distribution within each sample. As an aggregated measure, the median value for SMAPE and Aitchison's distances calculated on each sample was obtained. From now on, we then refer to these aggregated results simply as SMAPE and Aitchison's distance. As for other measures, results were aggregated outputs according to the used imputation method, reporting for each the mean and standard deviation of results of the corresponding pipelines. Based on these values, a rank was assigned for each metric to each tool increasingly with the worsening of the related performance. This allowed to observe (Tables 8.4 and 8.5) how in Simulation 1 *DrImpute* performed very well not only in recognizing which features got unobserved in the sequencing process, but also in retrieving correctly the information on relative count distribution within each feature, with a SMAPE around 7.7% for whichever normalization pairing and a mean Aitchison's distance of 18.7. As for the sparsity metric, *DrImpute* pipelines saw a drop in performance when applied to the other two Simulations (Tables 8.6-8.9)), however maintaining a good performance. It is noteworthy that the mean ranking for Aitchison's distance in Simulated Dataset 2 places DrImpute at the third position (even if the deviation from the first and second is very little), but when looking at detailed results (Table B.5, Appendix B) DrImpute results to be the best performing tool when combined with *edgeR* or *TSS* normalizations. Also *scImpute* acted very well in the first Simulated Dataset, but worse than all the pipelines involving the use of *DrImpute*. In this case, SMAPE was around 11.2% for all normalization pairings, with very little improvement in comparison of performing only normalization or no pre-processing (12.1%). Aitchison's measure performance was instead not optimal, giving a mean distance of 26.4 against the 19 of normalization pre-processing. However, in concordance with sparsity results, *scImpute* obtained the best results in Simulated Datasets 2 and 3 in terms of SMAPE metric, always showing an average performance when looking at Aitchison's distance. Regarding pipelines containing *LLSimpute*, *LowRank*, *zCompositions_SQ* and *zCompositions_CZM* imputations, they all performed poorly in terms of SMAPE in whichever of the three Simulations and independently on the chosen normalization step. The values obtained for this metric had a range of, respectively, 74.17-81.20%, 73.3-75.47%, 77.23-77.41% and 72.84-74.48% for Simulation 1; 77.37-83.18%, 69.47-70.64%, 71.11-72.39% and 68.94-72.33% for Simulation 2; 77.11-80.77%, 88.92-95.92%, 95.53-95.67% and 95.47-95.82% for Simulation 3 (Appendix B). Interestingly, *zCompositions*, in both the considered variants and in combination with several normalization approaches, had very good results in

terms of Aitchison's distance in all the Simulated Datasets. Regarding normalization-only pipelines, they usually performed better then *LLSimpute*, *LowRank*, *zCompositions_SQ* and *zCompositions_CZM* looking at both SMAPE and Aitchison's distance values, but always occupied average ranks, thus not being on the top preferable pre-processing approaches. A special case was observed for Simulated Dataset 3, where all the pipelines involving zero imputation resulted in some bias introduction when dealing with null values. In this example, raw and only-normalized data resulted the most adherent in terms of relative proportions to the real one, with a SMAPE of 2.72% and an Aitchison's distance of 4.97. To follow, *scImpute* and *DrImpute* pipelines obtained the best results in terms of SMAPE, while, again, *zCompositions* was the most reliable in terms of Aichison's distance.

The possible discrepancy between SMAPE and Aitchison's distace results could be explained in an excellent ability in recovering true relative abundances in the majority of features, that would result in a low mean SMAPE per sample (and, consequently, in a median overall SMAPE per dataset) joined with some sporadic case in which one or few errors of remarkable entity (and maybe in one/some highly abundant feature/s) were made in true feature values recovery. This would cause the Aichison's distance between the related true and pre-processed vectors to grow notably. The above results suggest (Simulation 1 and 2) that *scImpute* and *DrImpute* are reliable tools for true abundances recovery, with the former being affected by some considerable, sporadic error. This characteristic was found not to be present in *zCompositions* pipelines, that usually produced a reliable internal distribution (highlighted by Aitchison distance results) but a great SMAPE because of indiscriminate zero-imputation. *LLSimpute* and *LowRank* performance was found not to be appropriate for every simulated framework.

Table 8.4 Simulated Dataset 1, SMAPE between the ground truth and different pipelines. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is reported on the top-left of the Table.

	SMAPE		
Raw: 12.109	**Mean**	**SD**	**Rank**
DrImpute	7.665	0.244	1
scImpute	11.193	0.001	2
None	12.109	0.000	3
zCompositions_CZM	73.385	0.625	4
LowRank	74.204	0.804	5
zCompositions_SQ	77.299	0.064	6
LLSimpute	77.807	2.972	7

Table 8.5 Simulated Dataset 1, Aitchison's distance between the ground truth and different pipelines. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is reported on the top-left of the Table.

	Aitchison's distance		
Raw: 19.039	**Mean**	**SD**	**Rank**
DrImpute	18.663	0.164	1
None	19.039	0.000	2
zCompositions_SQ	19.125	0.057	3
zCompositions_CZM	20.922	1.631	4
scImpute	26.408	0.001	5
LowRank	60.699	12.881	6
LLSimpute	107.624	9.721	7

Table 8.6 Simulated Dataset 2, SMAPE between the ground truth and different pipelines. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is reported on the top-left of the Table.

	SMAPE		
Raw: 15.493	**Mean**	**SD**	**Rank**
scImpute	12.059	0.006	1
None	15.493	0.000	2
DrImpute	28.353	2.353	3
LowRank	70.097	0.441	4
zCompositions_CZM	70.161	1.457	5
zCompositions_SQ	71.682	0.550	6
LLSimpute	79.947	1.836	7

Table 8.7 Simulated Dataset 2, Aitchison's distance between the ground truth and different pipelines. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is reported on the top-left of the Table.

	Aitchison's distance		
Raw: 23.754	**Mean**	**SD**	**Rank**
zCompositions_SQ	23.363	0.177	1
None	23.754	0.000	2
DrImpute	24.289	2.271	3
scImpute	25.112	0.001	4
zCompositions_CZM	28.677	4.650	5
LowRank	44.262	12.244	6
LLSimpute	81.888	0.684	7

Table 8.8 Simulated Dataset 3, SMAPE between the ground truth and different pipelines. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is reported on the top-left of the Table.

	SMAPE		
Raw: 2.717	**Mean**	**SD**	**Rank**
None	2.717	0.000	1
scImpute	6.192	0.033	2
DrImpute	15.349	1.483	3
LLSimpute	78.801	1.183	4
LowRank	93.644	2.370	5
zCompositions_CZM	95.607	0.123	6
zCompositions_SQ	95.608	0.058	7

Table 8.9 Simulated Dataset 3, Aitchison's distance between the ground truth and different pipelines. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is reported on the top-left of the Table.

	Aitchison's distance		
Raw: 4.972	**Mean**	**SD**	**Rank**
None	4.972	0.000	1
zCompositions_CZM	7.940	2.177	2
zCompositions_SQ	12.362	1.754	3
DrImpute	19.376	2.990	4
scImpute	25.240	0.001	5
LLSimpute	55.788	3.687	6
LowRank	75.076	6.834	7

A summary of SMAPE and Aitchison's distance results for all the Simulated Datasets is reported in Figures 8.2 and 8.3.

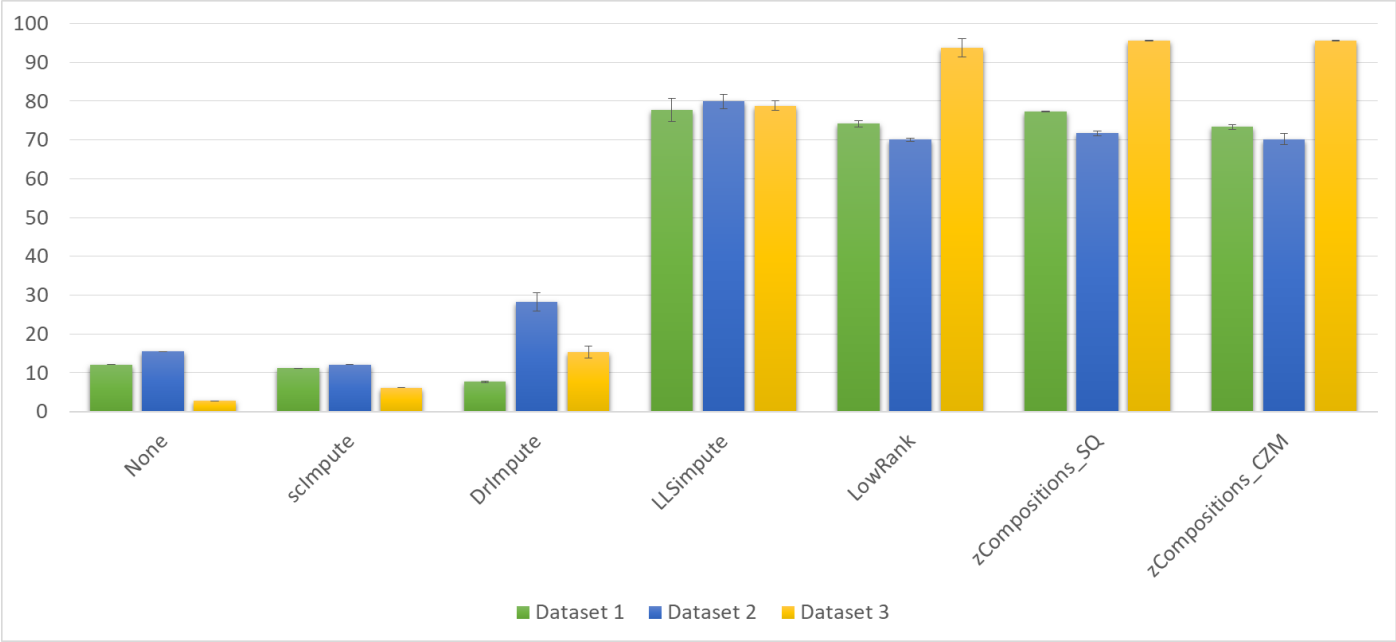


Fig. 8.2 SMAPE results in the different pipelines for the three Simulated Datasets. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

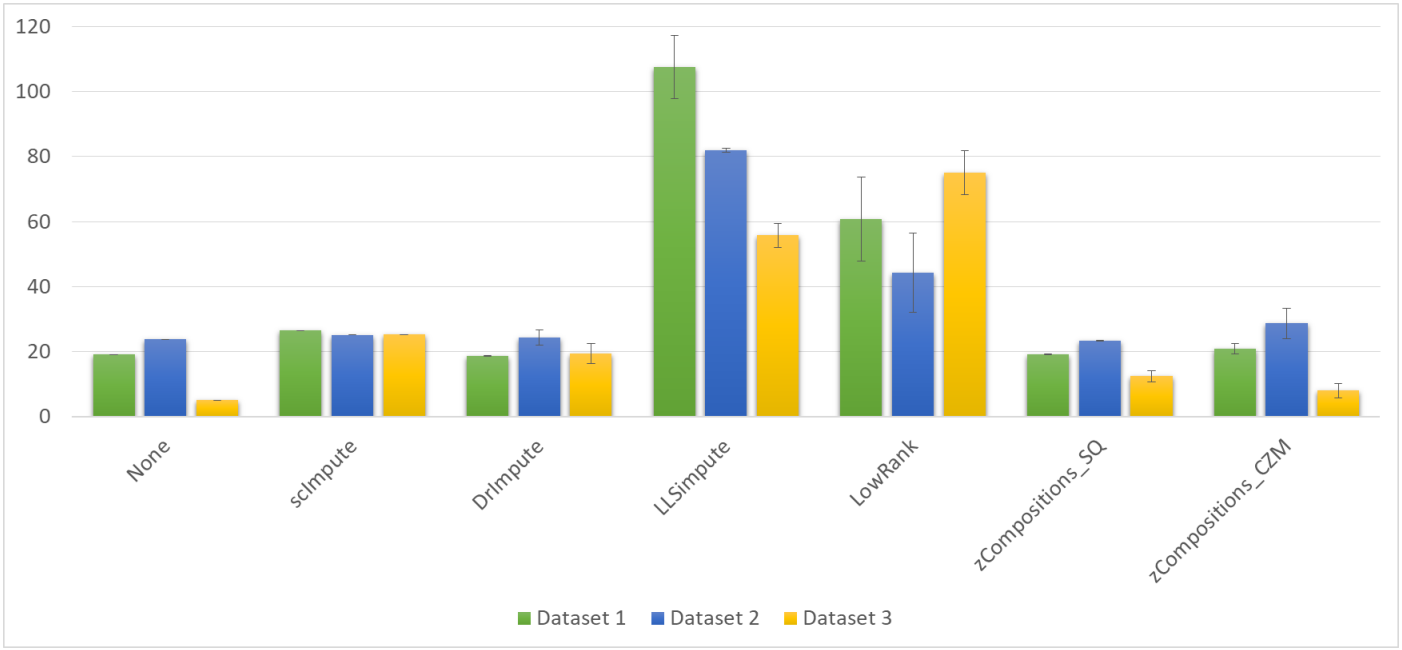


Fig. 8.3 Aitchison's distance results in the different pipelines for the three Simulated Datasets. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

8.3 Alpha diversity

One of the most important questions about metagenomic data is about sample species internal composition. As introduced in Chapter 7, in microbiome studies this issue is commonly answered by the calculation of alpha diversity indices. This permits to have an overview of sample composition, in terms of both the number of detected species (*richness*) and the distribution of count within them (*evenness*), i.e. their proportional abundance distribution. Alpha diversity values are used in real data analysis to detect differences in composition between the groups contained in the experiment (and into the related count matrix). This is usually achieved by performing group-group statistical tests to compare the related alpha values distributions. Consequently, we decided to evaluate the impact of different pipelines application by looking at the consequences on this statistical testing procedure. In particular, for each alpha index considered in this thesis, we calculated the number of correct group-group comparisons in relation to the results obtained performing the same tests in the ground truth. This was obtained by using a one-sided non parametric Mann-Whitney test, to detect both statistical difference and the direction of change between compared groups. Additionally, for the richness index, the mean absolute error (MAE) and SMAPE between the ground truth and datasets obtained from different pipelines were calculated, to join the information on the goodness in recovering lost information (unobserved OTUs) and the impact of this recovery when performing comparisons between groups. Results were then expressed for each pipeline in terms of percentage of comparisons that showed discordance with the ground truth. As for other metrics, results are reported in an aggregated way within this chapter and in detail in Appendix B.

Results on the calculation of MAE and SMAPE on richness index showed (Figures B.1-B.6, Appendix B) how in raw data detected species were notably less then the number of species present in real data (MAE = 316.92 and SMAPE = 15.54% for Simulation 1, MAE = 85.66 and SMAPE = 15.7% for Simulation 2, MAE = 35.05 and SMAPE = 19.88% for Simulation 3). As normalization methods only act as scaling procedures, no improvement was obtained in data in which only this pre-processing step was performed. When imputation was applied, species recovery was performed and a change in Richness index was observed. The sparsity results seen in Tables 8.1-8.3 explain why *scImpute* pipelines were found to be the best pre-processing approaches in recreating samples richness, followed by *DrImpute*, in combination with whichever normalization technique. As for other metrics, Simulation 3 showed a slight performance worsening for *DrImpute*, in concordance with sparsity underestimation. Pipelines including *LLSimpute*, *LowRank* and *zCompositions* imputations obtained, as for sparsity, the worst performance in all the three Simulated Datasets.

Results obtained from Mann-Whitney tests reported in Table 8.10 show how high performance in rare species recovery are reflected in a low number of incorrect (in relation to the ground truth) group-group comparisons. In fact, from Table 8.10 and Figure 8.4 it's evident the effect of *scImpute* pipelines application in increasing richness comparisons results in all the three Simulations. Also *DrImpute* pipelines confirmed their good behaviour in Simulation 1, but it saw a drop on performance in the other two datasets. All other tool confirmed their bad performance also according to this metric.

Table 8.10 Results on richness index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is also reported for each dataset.

RICHNESS			
Dataset 1			
Raw: 37.36%	**Mean**	**SD**	**Rank**
DrImpute	2.35%	0.42%	1
scImpute	23.08%	0.00%	2
None	37.36%	0.00%	3
LLSimpute	42.07%	13.90%	4
LowRank	79.28%	10.26%	5
zCompositions_SQ	100.00%	0.00%	6
zCompositions_CZM	100.00%	0.00%	7
Dataset 2			
Raw: 39.29%	**Mean**	**SD**	**Rank**
scImpute	10.71%	0.00%	1
None	39.29%	0.00%	2
LLSimpute	58.67%	8.21%	3
DrImpute	64.29%	0.00%	4
LowRank	81.63%	10.99%	5
zCompositions_SQ	92.86%	0.00%	6
zCompositions_CZM	92.86%	0.00%	7
Dataset 3			
Raw: 6.06%	**Mean**	**SD**	**Rank**
None	6.06%	0.00%	1
scImpute	6.06%	0.00%	2
DrImpute	32.68%	3.70%	3
LowRank	82.47%	4.50%	4
LLSimpute	91.13%	7.72%	5
zCompositions_SQ	95.45%	0.00%	6
zCompositions_CZM	95.45%	0.00%	7

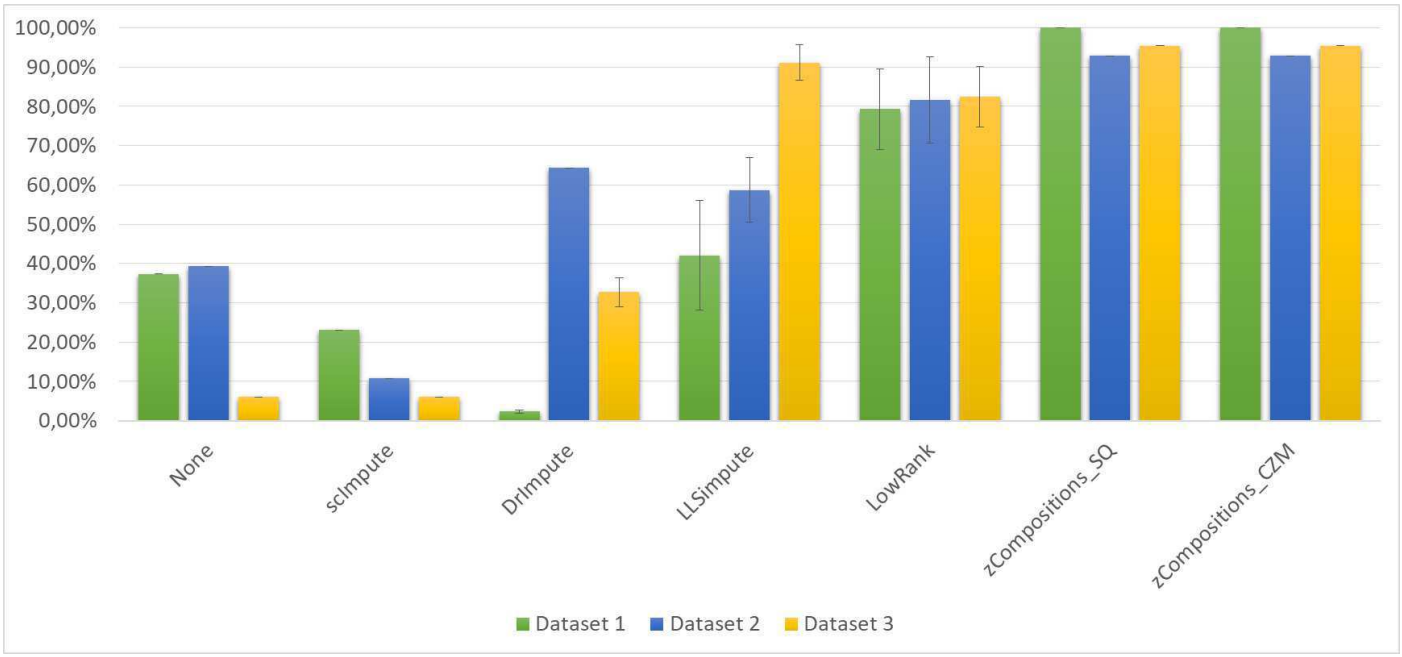


Fig. 8.4 Results on richness index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

Regarding alpha diversity measured as evenness, the worst performance on Pielou index values was obtained in all tested frameworks by *LLSimpute*, both applied singularly and preceded by a normalization step (Table 8.11 and Figure 8.5). Little or not appreciable difference with raw data results was observed in pipelines involving *LowRank*, while a slight worsening was observed for *zCompositions* pipelines in both the configurations and in association with all the normalizations. The best performance was again obtained by *DrImpute*, followed by *scImpute*, in Simulated Dataset 1, while the trend is inverted in the other two Simulations. Results were, however, always comparable between pipelines involving these two imputation approaches.

Table 8.11 Results on Pielou index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is also reported for each dataset.

PIELOU			
Dataset 1			
Raw: 9.89%	**Mean**	**SD**	**Rank**
DrImpute	9.89%	0.00%	1
scImpute	5.49%	0.00%	2
None	3.77%	2.53%	3
LLSimpute	57.14%	5.92%	4
LowRank	11.62%	2.53%	5
zCompositions_SQ	13.19%	0.00%	6
zCompositions_CZM	13.66%	0.59%	7
Dataset 2			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
scImpute	0.00%	0.00%	2
zCompositions_SQ	3.57%	0.00%	3
zCompositions_CZM	3.57%	0.00%	4
LowRank	6.63%	5.62%	5
DrImpute	13.78%	3.21%	6
LLSimpute	64.80%	12.10%	7
Dataset 3			
Raw: 4.55%	**Mean**	**SD**	**Rank**
None	4.55%	0.00%	1
scImpute	17.97%	1.05%	2
LowRank	31.82%	6.25%	3
DrImpute	36.15%	2.04%	4
zCompositions_CZM	39.18%	0.57%	5
zCompositions_SQ	39.39%	0.00%	6
LLSimpute	66.45%	8.55%	7

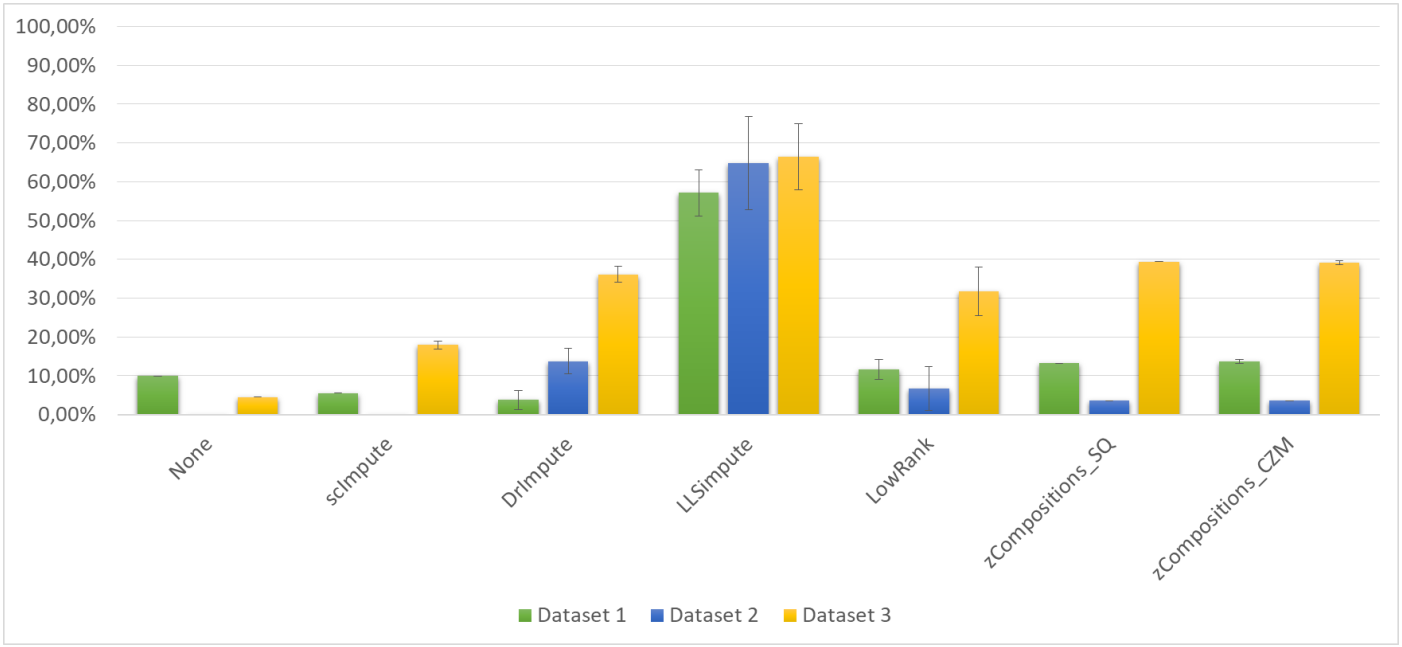


Fig. 8.5 Results on Pielou index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

In Tables 8.12-8.14 are reported results about the three diversity indices shown in Chapter 7 (Tail, Shannon and inverse Simpson). As for all other metrics, *LLSimpute* and *LowRank* were the overall worst performing pipelines in all the three diversity measures and Simulations (Figures 8.6-8.8). On the contrary, *scImpute* and *DrImpute* pipelines were found to be the most-effective approaches in reconstructing low-abundance features, as suggested by the good performance in Tail index. In fact, they always obtained low percentages of wrong (compared to ground truth results) group-group comparison outputs, with the unique exception of the pipeline composed by *GMPR* and *DrImpute* that had a 25% of error in Simulation 2 (see details in Appendix B). Also for *zCompositions_SQ* results obtained for Tail metric were good, whereas *zCompositions_CZM* had very poor performance.

Looking at the other two metrics, i.e. Shannon and inverse Simpson (iSimpson) indices, *zCompositions*, *scImpute* and *DrImpute* had very similar overall behaviours in the first and second Simulations, whereas in the third Simulation *scImpute* pipelines showed a decrease in performance. As described in Chapter 7, Shannon and iSimpson indices are related to information on averagely and highly abundant features, respectively. *scImpute* slight decrease in goodness of results could mean that the related pipelines tend to impute some zeros with a too high value, thus slightly changing the population profile inserting at mid or high values there were not present in the ground truth. Also for these indices, *LLSimpute* and *LowRank* were very poor and very distant from all other pipelines results (Figures 8.6-8.8).

Table 8.12 Results on Shannon index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is also reported for each dataset.

SHANNON			
Dataset 1			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
zCompositions_SQ	0.00%	0.00%	2
zCompositions_CZM	0.47%	0.59%	3
scImpute	1.10%	0.00%	4
DrImpute	3.14%	1.17%	5
LowRank	7.54%	3.26%	6
LLSimpute	58.40%	7.11%	7
Dataset 2			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
zCompositions_SQ	0.00%	0.00%	2
zCompositions_CZM	0.00%	0.00%	3
DrImpute	2.04%	2.81%	4
LowRank	4.08%	5.62%	5
scImpute	7.14%	0.00%	6
LLSimpute	60.71%	10.71%	7
Dataset 3			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
zCompositions_SQ	0.00%	0.00%	2
zCompositions_CZM	0.65%	0.81%	3
DrImpute	2.38%	1.72%	4
scImpute	18.18%	0.00%	5
LowRank	41.13%	16.07%	6
LLSimpute	65.80%	18.33%	7

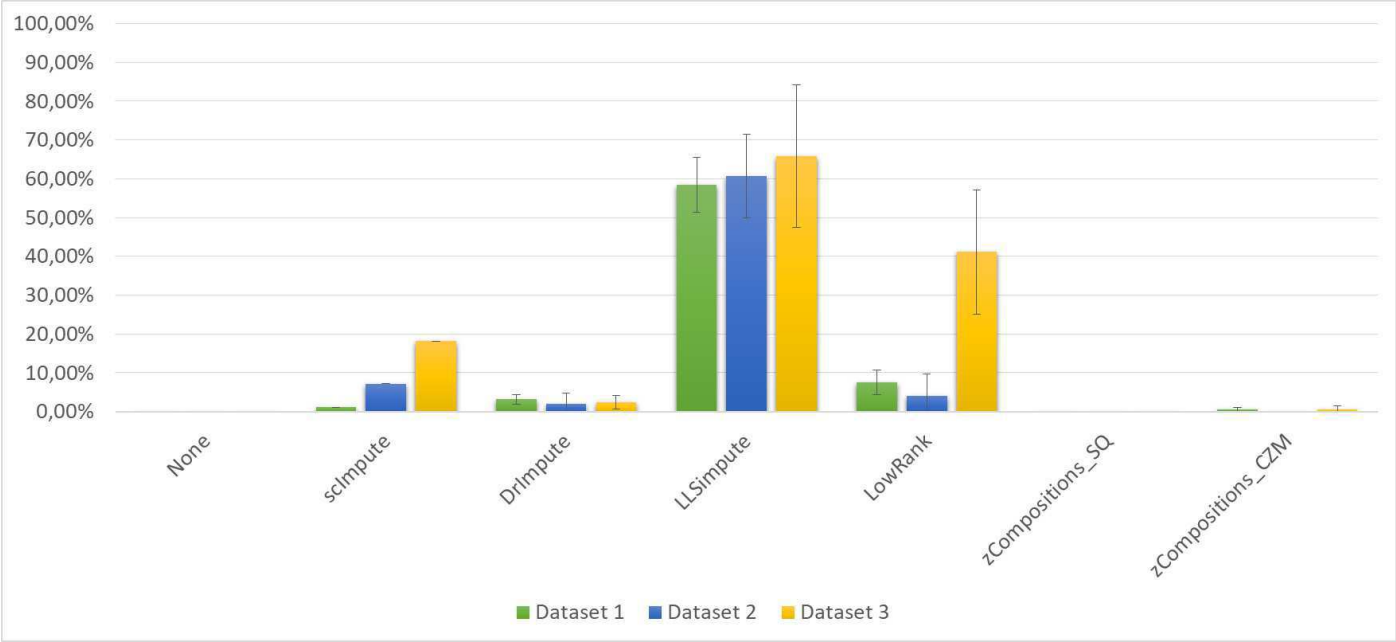


Fig. 8.6 Results on Shannon index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

Table 8.13 Results on iSimpson index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is also reported for each dataset.

ISIMPSON			
Dataset 1			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
DrImpute	0.00%	0.00%	2
zCompositions_SQ	0.00%	0.00%	3
zCompositions_CZM	0.00%	0.00%	4
scImpute	1.10%	0.00%	5
LowRank	7.54%	4.55%	6
LLSimpute	56.99%	5.29%	7
Dataset 2			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
scImpute	0.00%	0.00%	2
DrImpute	0.00%	0.00%	3
zCompositions_SQ	0.00%	0.00%	4
zCompositions_CZM	0.00%	0.00%	5
LowRank	2.04%	4.05%	6
LLSimpute	58.16%	12.99%	7
Dataset 3			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
zCompositions_SQ	0.00%	0.00%	2
zCompositions_CZM	0.00%	0.00%	3
DrImpute	0.87%	0.81%	4
scImpute	15.15%	0.00%	5
LowRank	21.21%	16.39%	6
LLSimpute	61.69%	8.56%	7

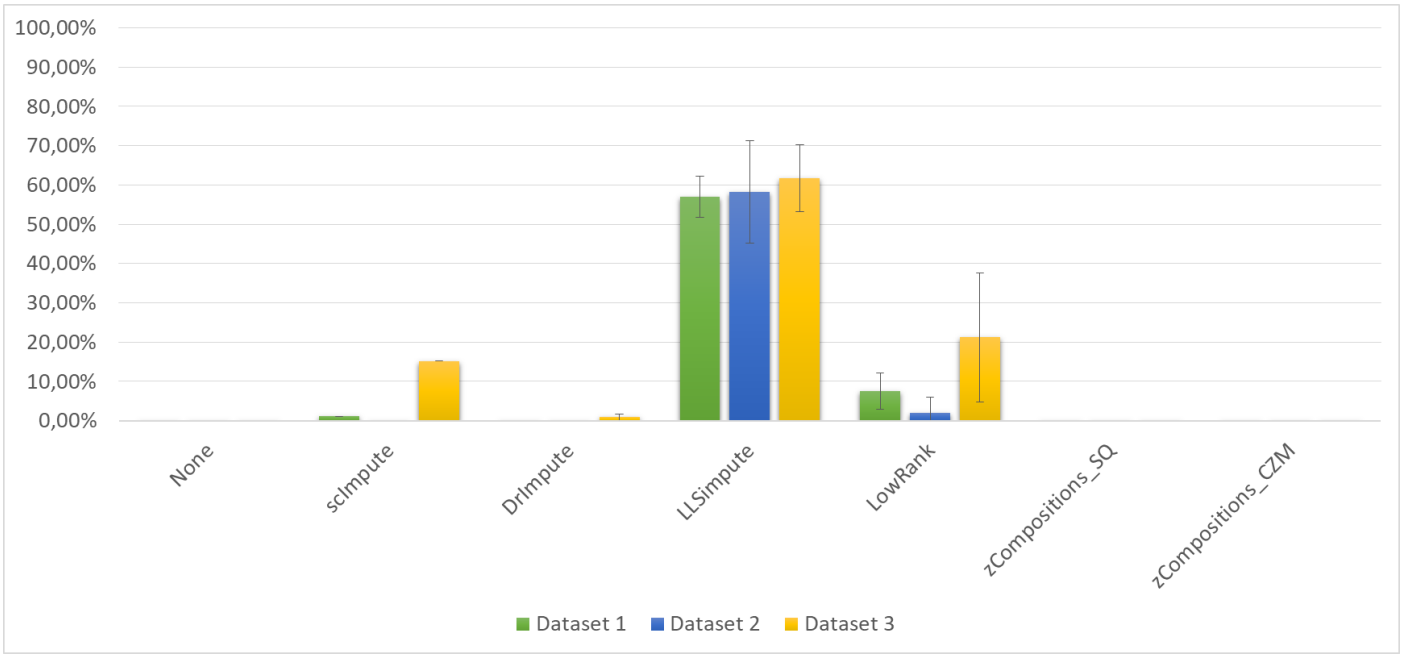


Fig. 8.7 Results on iSimpson index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

Table 8.14 Results on Tail index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are ordered according to performance ranking. Pre-processed datasets results were aggregated according to the zero-imputation method included in pipeline; for each, the mean and standard deviation over different normalizations are shown. Raw data result is also reported for each dataset.

TAIL			
Dataset 1			
Raw: 4.40%	**Mean**	**SD**	**Rank**
DrImpute	3.92%	1.40%	1
zCompositions_SQ	4.24%	0.42%	2
None	4.40%	0.00%	3
scImpute	4.40%	0.00%	4
zCompositions_CZM	15.07%	7.45%	5
LowRank	37.36%	14.56%	6
LLSimpute	54.95%	12.29%	7
Dataset 2			
Raw: 10.71%	**Mean**	**SD**	**Rank**
scImpute	7.14%	0.00%	1
None	10.71%	0.00%	2
zCompositions_SQ	14.29%	0.00%	3
DrImpute	14.80%	9.32%	4
zCompositions_CZM	21.94%	8.85%	5
LowRank	38.27%	30.21%	6
LLSimpute	70.41%	14.39%	7
Dataset 3			
Raw: 0%	**Mean**	**SD**	**Rank**
None	0.00%	0.00%	1
zCompositions_SQ	4.11%	4.08%	2
DrImpute	6.28%	2.82%	3
scImpute	9.09%	0.00%	4
zCompositions_CZM	41.34%	24.71%	5
LLSimpute	70.13%	17.86%	6
LowRank	79.00%	6.97%	7

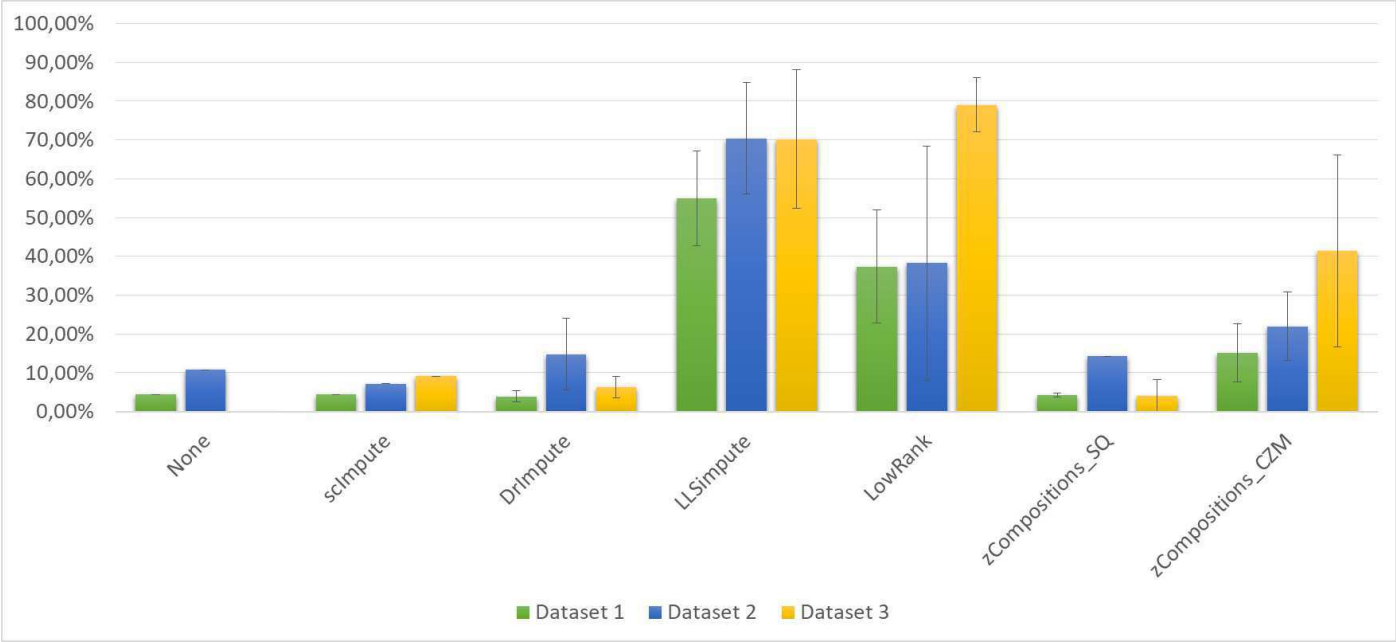


Fig. 8.8 Results on Tail index for the three Simulated Datasets in terms of percentage of group-group comparisons not agreeing with the ground truth. Results are aggregated according to the imputation method used in the pipelines and are reported as mean values calculated among normalization methods. The error bars represent the related standard deviations.

In conclusion, *scImpute*-, *DrImpute*- and *zCompositions_SQ*-based pipelines resulted to be the best performing approaches when looking at alpha diversity measures. The above observations on internal abundance proportions conservation were mitigated by these results, letting suppose that the sporadic errors that caused bad performance in Aitchison's distance metric were not dramatic enough to cause a change in alpha diversity results and conclusions.

8.4 Beta diversity

Even if alpha indices are very useful to look at each sample characteristics, they give no information about between-sample relations. As explained in Chapter 7, in 16S sequencing studies this aim is achieved by calculating beta diversity indices. This values are then used to measure dissimilarity between samples, in order to collect the ones that resemble to each other and divide the whole into different groups. With this aim, two types of beta analyses were performed: one based on abundance information (Bray-Curtis dissimilarity) and one based on presence/absence data (Whittaker index), showing different aspects of the considered matrix. The first dissimilarity was used to build a distance matrix on which NMDS dimensionality reduction was performed.

In Figures 8.9-8.11 are presented the results obtained for NMDS dimensionality reduction on Bray-Curtis distance performed on the threes Simulated Datasets. Looking at the first one (Figure 8.9), it can be observed that in the real dataset samples belonging to the same group (biological replicates) tended to be very nearly located in the two dimensional NMDS subspace. This characteristic got lost after sequencing process, i.e. in raw data, where samples of the same group tended to move away from each other and to form greater spacial clusters with members of other groups. Normalization-only pipelines were unable to recover the original structure of data, thus showing very little difference in spacial disposition from raw data. The best results in this sense were obtained by *DrImpute* and *scImpute* pipelines including previous normalization. In fact, these pipelines were able to rejoin samples belonging to the same experimental condition and divide different groups. Remarkably, the use of *GMPR* normalization specifically designed for zero-inflated count data prior to *DrImpute* and *scImpute* imputation allowed to obtain the spatial configuration that most resembled the ground truth. On the contrary, pipelines involving *LLSimpute*, *LowRank* and *zCompositions* generally brought additional noise, favouring the scattering of observations on the plain and consequently leading to group information loss.

Regarding Simulation 2 (Figure 8.10), the best results were obtained by *scImpute* pipelines including normalization; *DrImpute* pipelines had also acceptable results, but in many cases they tended to flatten data and to cancel also intra-group variation. These

pipelines were able to rejoin samples belonging to the same experimental condition and divide different groups. As for the first analyzed dataset, pipelines involving *LLSimpute*, *LowRank* and *zCompositions* generally brought additional noise, implying the scattering of observations and leading to group information loss.

In Simulation 3 (figure 8.11), we recall very little information got lost during the sequencing process. This reflected on NMDS results on Bray-Curtis distance, from which we can see (Figure 8.11) that real, raw and normalized data are very similar one to each other. This dataset was included in this study with the principal aim of assessing the possible bias introduction of zero-imputation pipelines in experiments where no o little sequencing zeros are present. NMDS plots showed the best performance (excluding normalizatio-only pipelines) in this sense is achieved by *scImpute* pipelines, that even if underestimating sparsity gave the most adherent results to real data. Also *DrImpute* performance was good, even though it tended to impute also a part of biological zeros thus causing an artificial over-separation of groups.

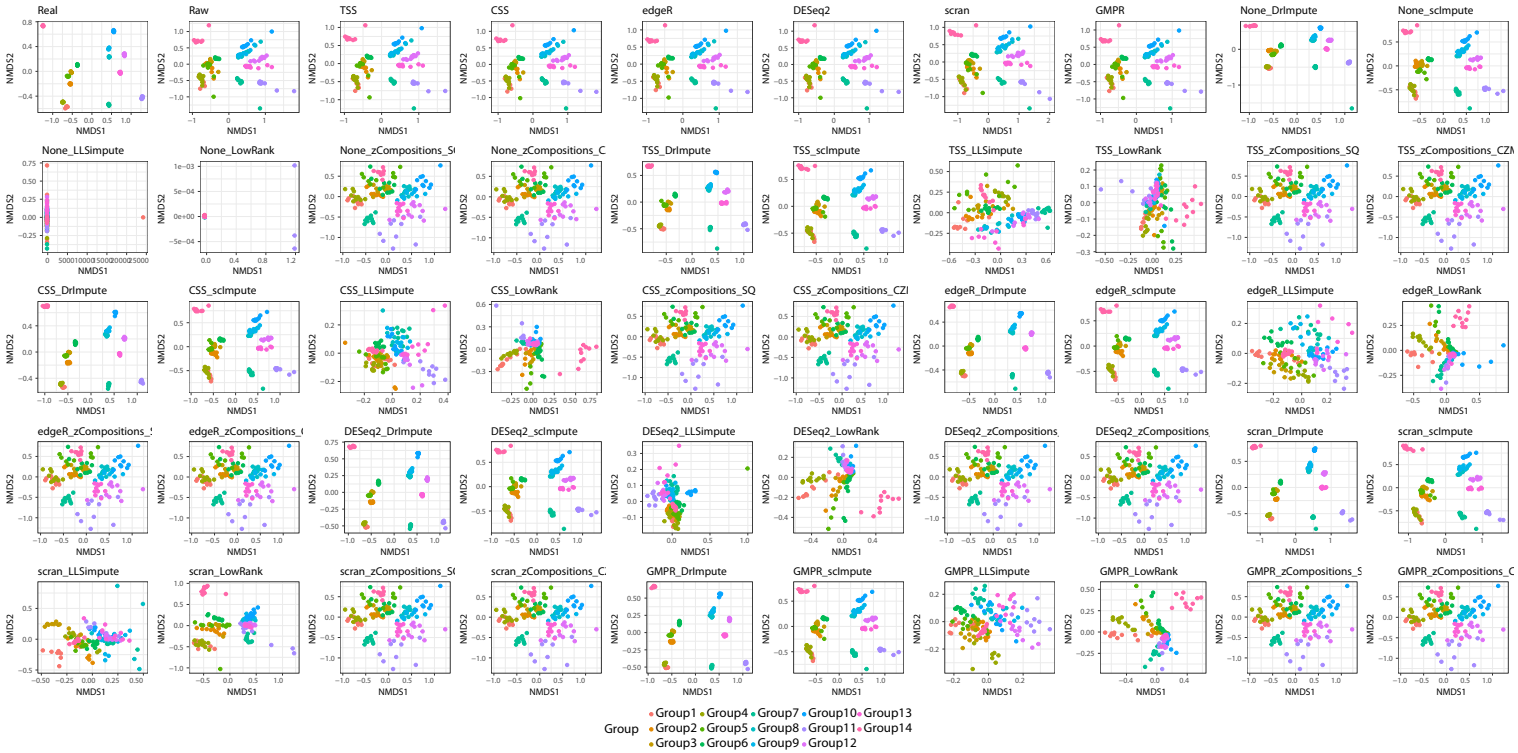


Fig. 8.9 Simulated Dataset 1 - NMDS plot on real, raw and pre-processed data.

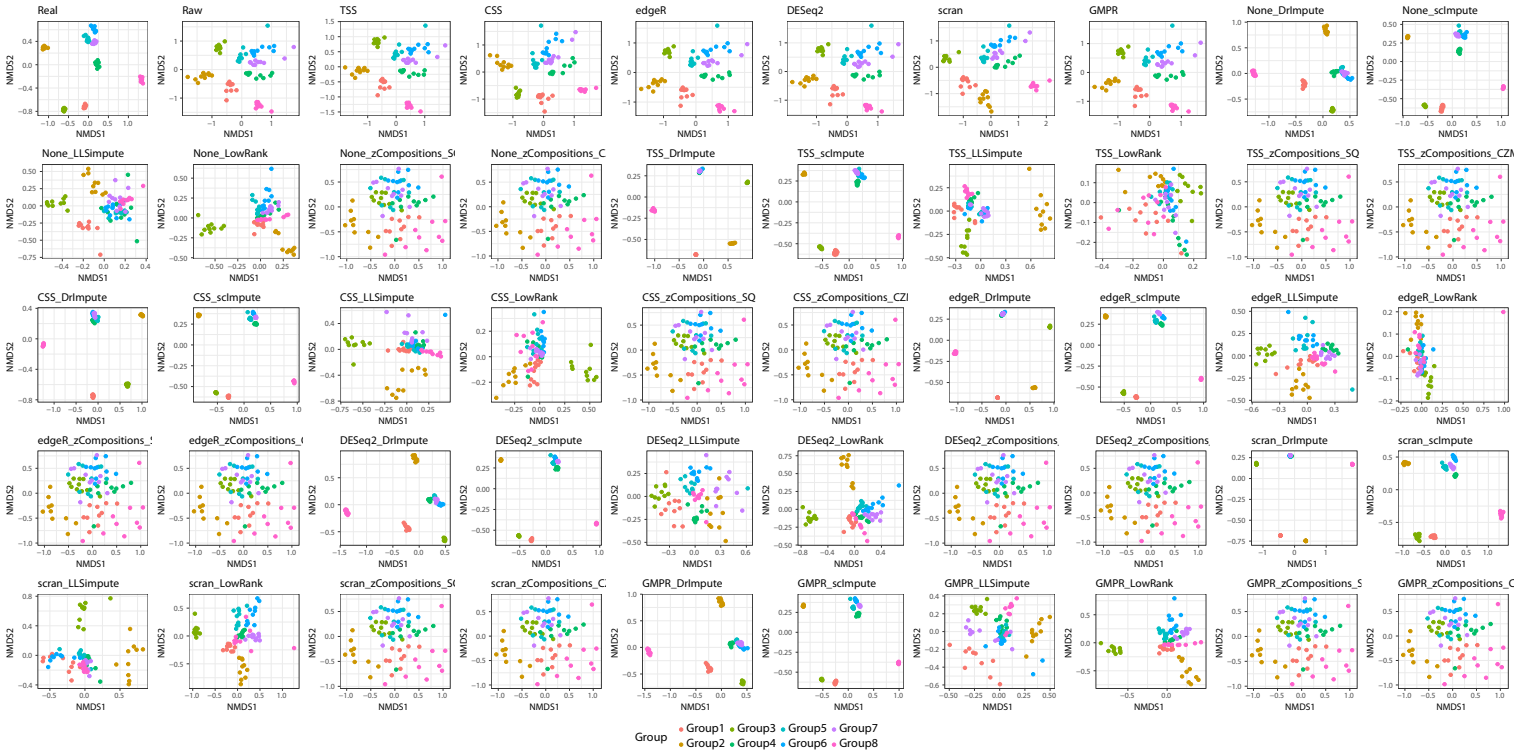


Fig. 8.10 Simulated Dataset 2 - NMDS plot on real, raw and pre-processed data.

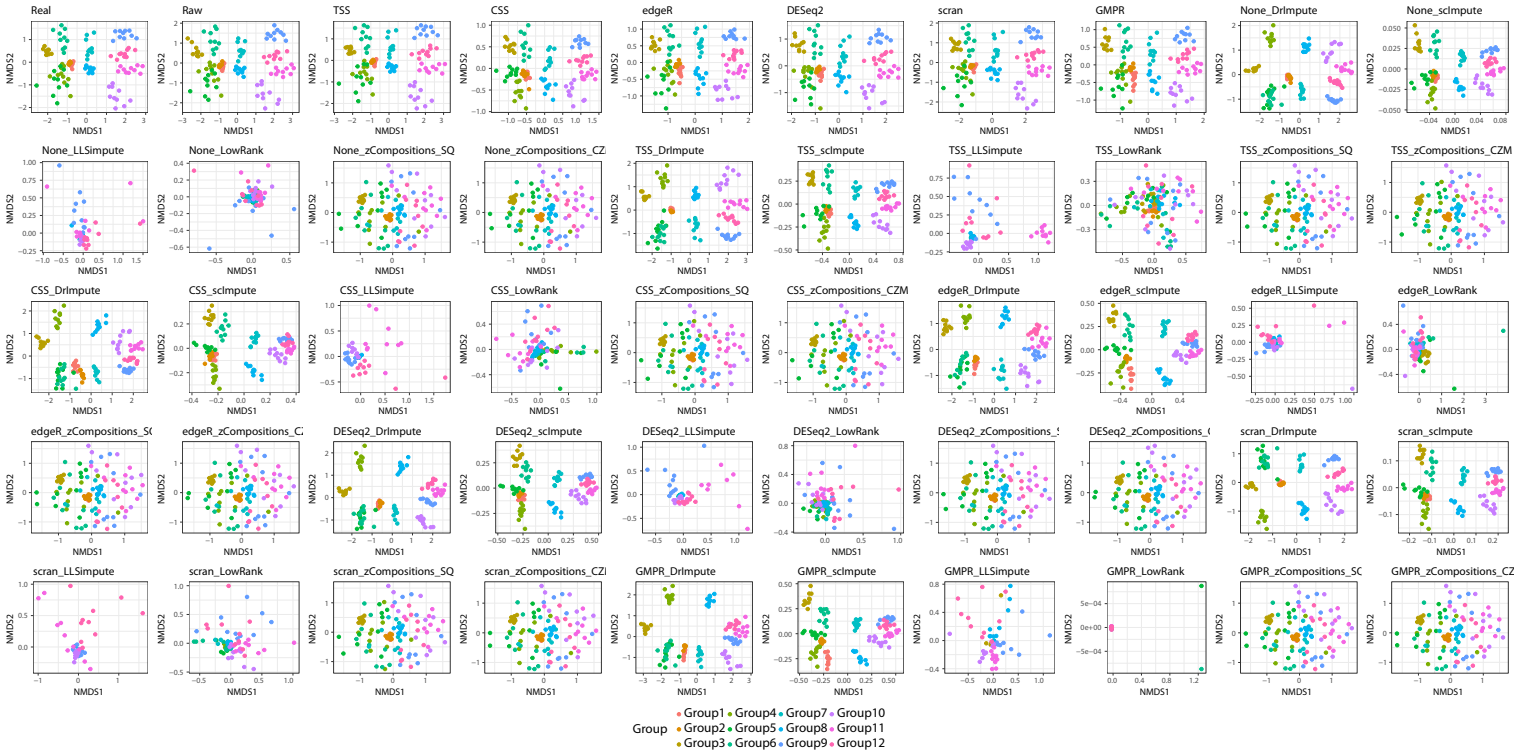


Fig. 8.11 Simulated Dataset 3 - NMDS plot on real, raw and pre-processed data.

Beta index on presence/absence information was used to calculate distances based on the number of shared and exclusive features between samples. The obtained distance matrices were represented as heatmaps in which a colour scale links the shade to a distance value (Figure 8.12). The results based on this metric greatly mimicked the ones obtained considering abundance data, letting think samples grouping is mainly based on richness information, i.e. on the presence/absence of features. For Simulation 1, pipelines involving the use of *DrImpute* and *scImpute* imputations were found to be the most effective in recreating real distances (in terms of Witthaker index) among samples. Normalization step had generally no improvement on *scImpute* results, except for *GMPR* normalization. On the contrary, *DrImpute* benefited from any prior normalization, with best results obtained in association with *GMPR* normalization. One difference between *DrImpute* and *scImpute* results is that the first one tends to have a "block" behaviour, meaning that all members of a group tend to have the same distance between all the members of another group, while in *scImpute* values also differences in distance between samples belonging to the same group, characteristic that is not observable in the ground truth.

Also results based on presence/absence information available from Simulated Dataset 2 confirmed the ones obtained considering abundance data. Also in this context, pipelines involving the use of *scImpute* imputations were found to be the most appropriate for information retrieval. In fact, Whittaker beta diversity heatmaps (8.13) clearly showed an almost perfect information recovery, both looking at intra-group and inter-group distances. Normalization step had generally visible improvement on *scImpute* results, despite no prior normalization is asked to the user from the tool. *DrImpute* over-imputation tendency had clear consequences in this metric. In fact, it is clearly visible a decrease in the third, fourth and fifth groups distances, with the consequent creation of a unique, big group including them all.

For Simulation 3 (Figure 8.14) results also reflected the abundance-based analysis, confirming that the most realistic results were obtained for pipelines that included *scImpute* imputation.

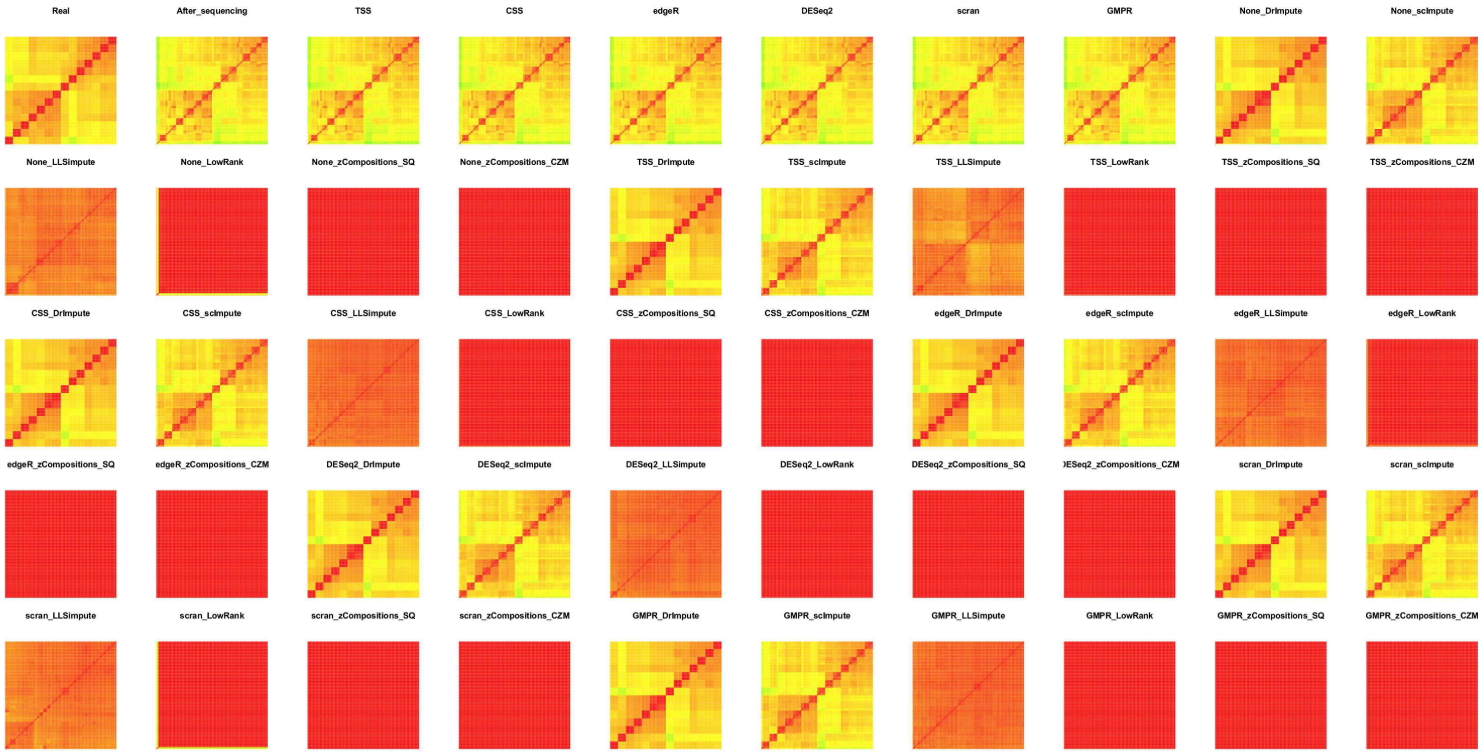


Fig. 8.12 Simulated Dataset 1 - Heatmap of Whittaker beta diversity calculated on Real, Raw and pre-processed data.

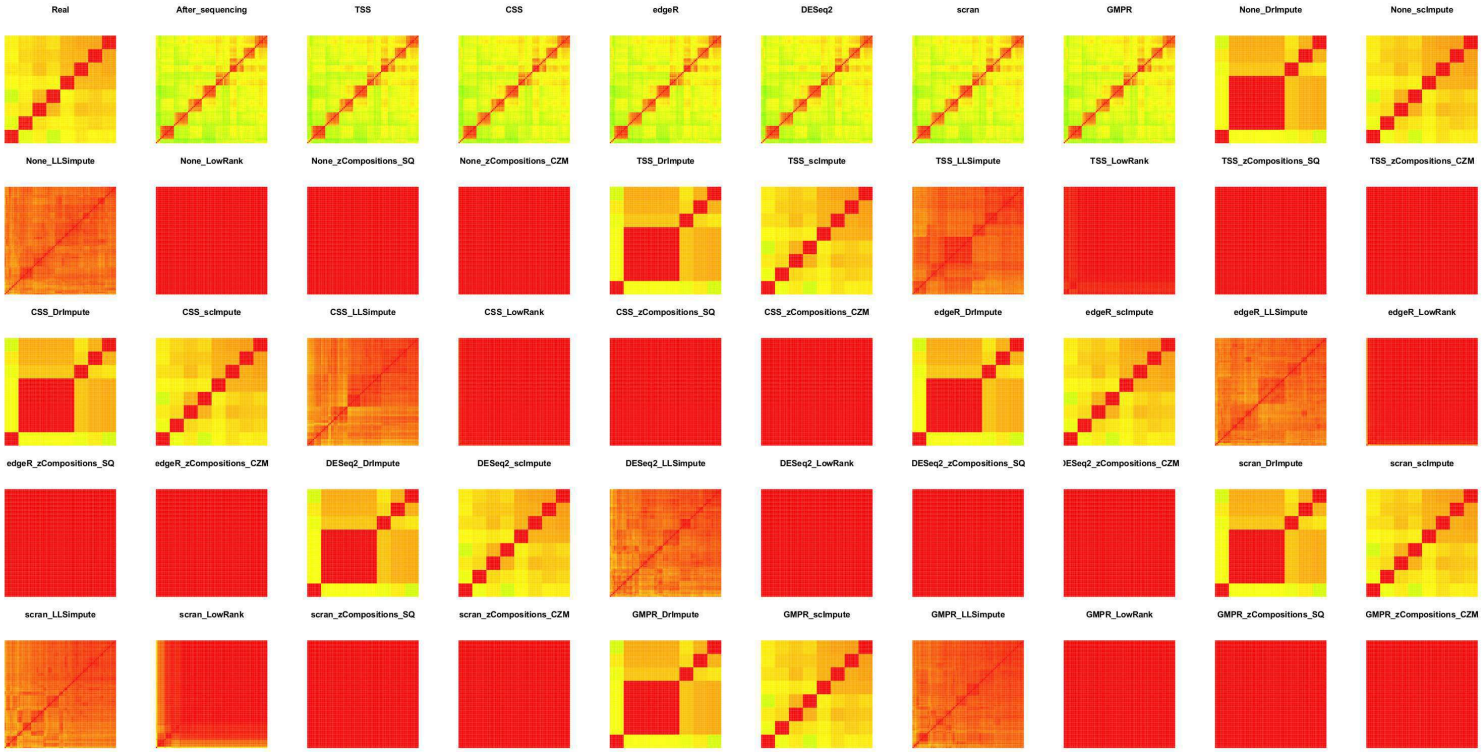


Fig. 8.13 Simulated Dataset 2 - Heatmap of Whittaker beta diversity calculated on Real, Raw and pre-processed data.

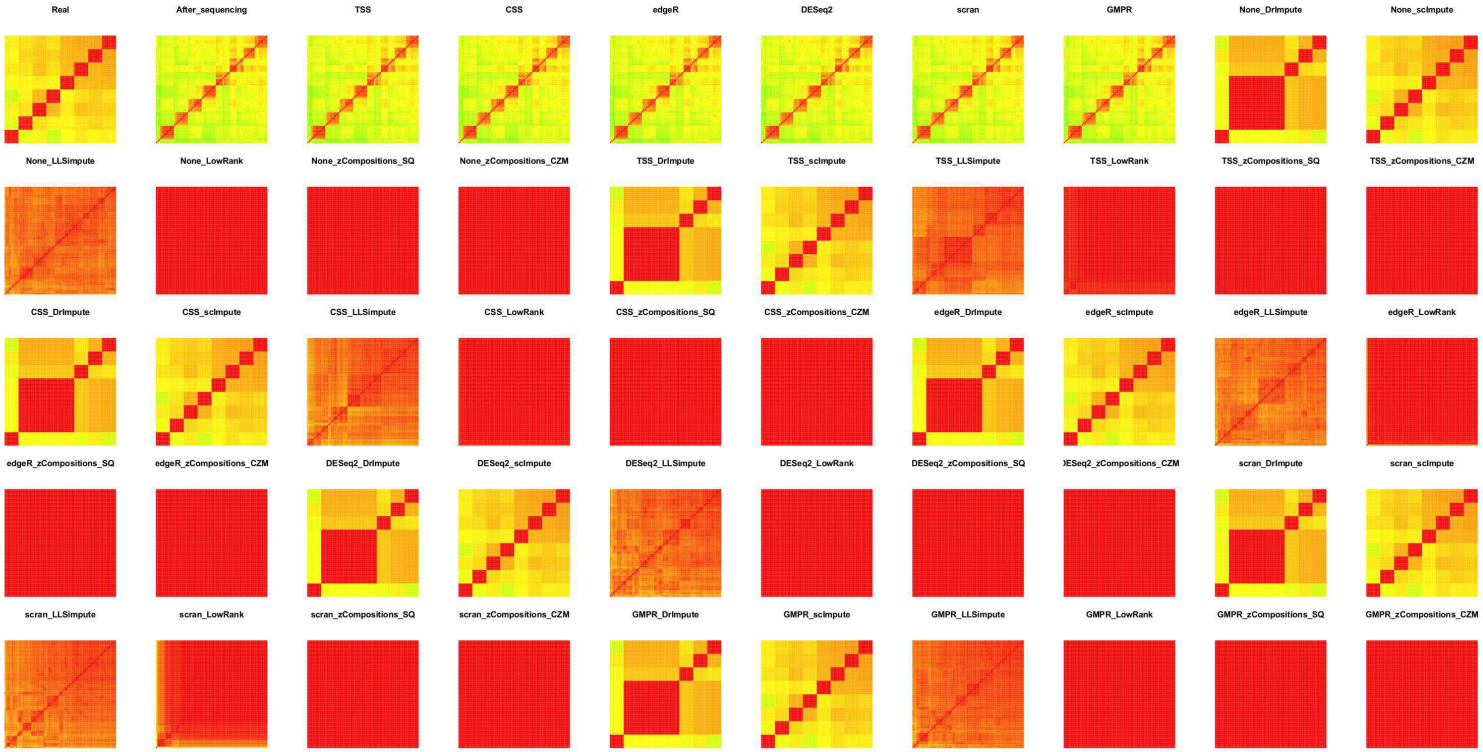


Fig. 8.14 Simulated Dataset 3 - Heatmap of Whittaker beta diversity calculated on Real, Raw and pre-processed data.

8.5 Differential abundance analysis

As introduced in the previous chapter, a test for change in differential abundance analysis results was performed as a further measure on each pipeline ability to recover original data structure. The result of this investigation was summarized into horizontal box plots, with the median of Jaccard indices obtained from comparisons between raw and real data highlighted by a dashed vertical line. The labels separated by a comma potentially present on the right of each box plot are related to the significance of the Mann Whitney test performed to test for improvement respect to raw data results and, in case of significant difference ($p<0.05$), to the magnitude of the size effect, respectively. From the plot related to Simulated Dataset 1 results (Figure 8.15) it is immediately visible a great improvement in recreating true differential analysis results for the pipelines involving *DrImpute* and *scImpute* imputations, both enhancing their performance when data were pre-normalized. In fact, both *scImpute* and *DrImpute* combined with whichever normalization maintained their significance in improvement respect to no pre-processing but changed their effect sizes from "Medium" (M) and "Very Large" (VL) to "Large" (L) and "Huge" (H) respectively, except in the case of pre-normalitaion with *scran*, where no improvement was seen for *DrImpute* and a worsening was observable for *scImpute*. It is also noteworthy that also normalization-only pre-processing led to a general improvement in retrieving differential abundant features compared to the performance on raw data (except from *scran* normalization), but the related effect sizes were all found to be "Very Small" (VS) or "Small" (S), thus indicating a non-null but very slight gain in using them.

The same analysis performed on Simulation 2 (Figure 8.16) produced the best results in recreating true differential analysis results for the pipelines involving *scImpute* imputations, that had significant improvements over raw data and usually related "Large" (L) effect sizes, except for the combination with *scran* normalization, where no improvement was observed. Contrarily, *DrImpute* introduction in pre-processing brought a worsening in performance, both with and without prior normalization. This could be linked to the over-imputation behaviour highlighted in the previous sections, that contributed to the variation or creation of differentially abundant features between the groups. It is also noteworthy that, also for this dataset, normalization-only pre-processing led to a general improvement in retrieving real differential abundant features, but again the related effect sizes were all found to be "Very Small" (VS) or "Small" (S).

Regarding the last Simulation, *DrImpute* and *scImpute* showed (Figure 8.17) a great improvement in differential analysis results compared to using raw or normalized data. Even if the pipelines that include this two tools tended to correct also for a little part of biological zeros, the benefit brought by correctly recovered information was definitely higher than the

bias introduced by zero over-replacement. This is probably due to the fact that the majority of "false" corrections did not affect differences between groups when looking at each group population, being the newly introduced values in concordance with the already (very few) present among biological replicates.

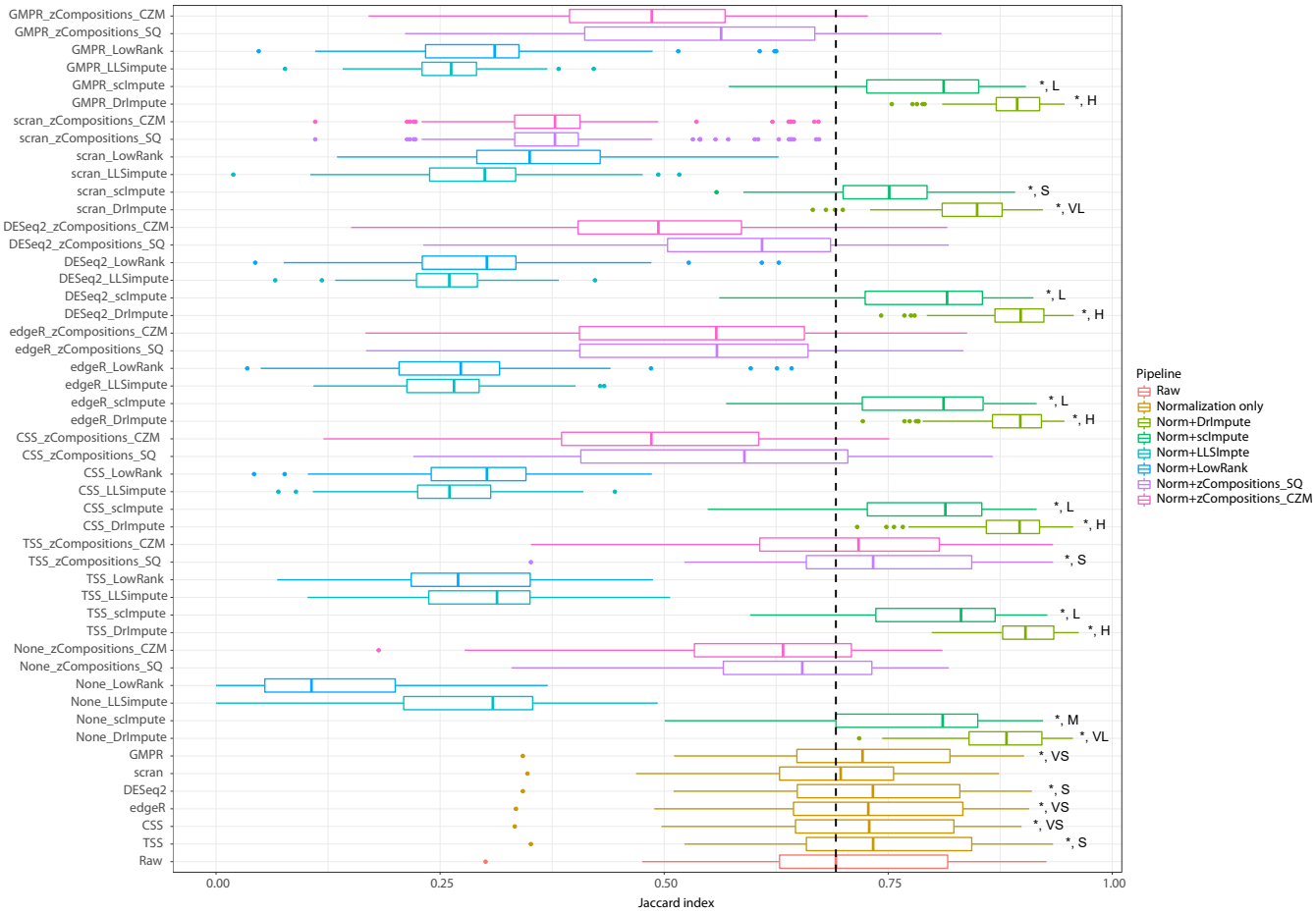


Fig. 8.15 Simulated Dataset 1 - Horizontal box plots of Jaccard index results obtained from comparisons between real and raw/pre-processed data. The median of values on raw data is highlighted by a dashed vertical line. The two labels separated by a comma on the right of each box plot indicate if significance was found in relation to raw data and the magnitude of the size effect, respectively.

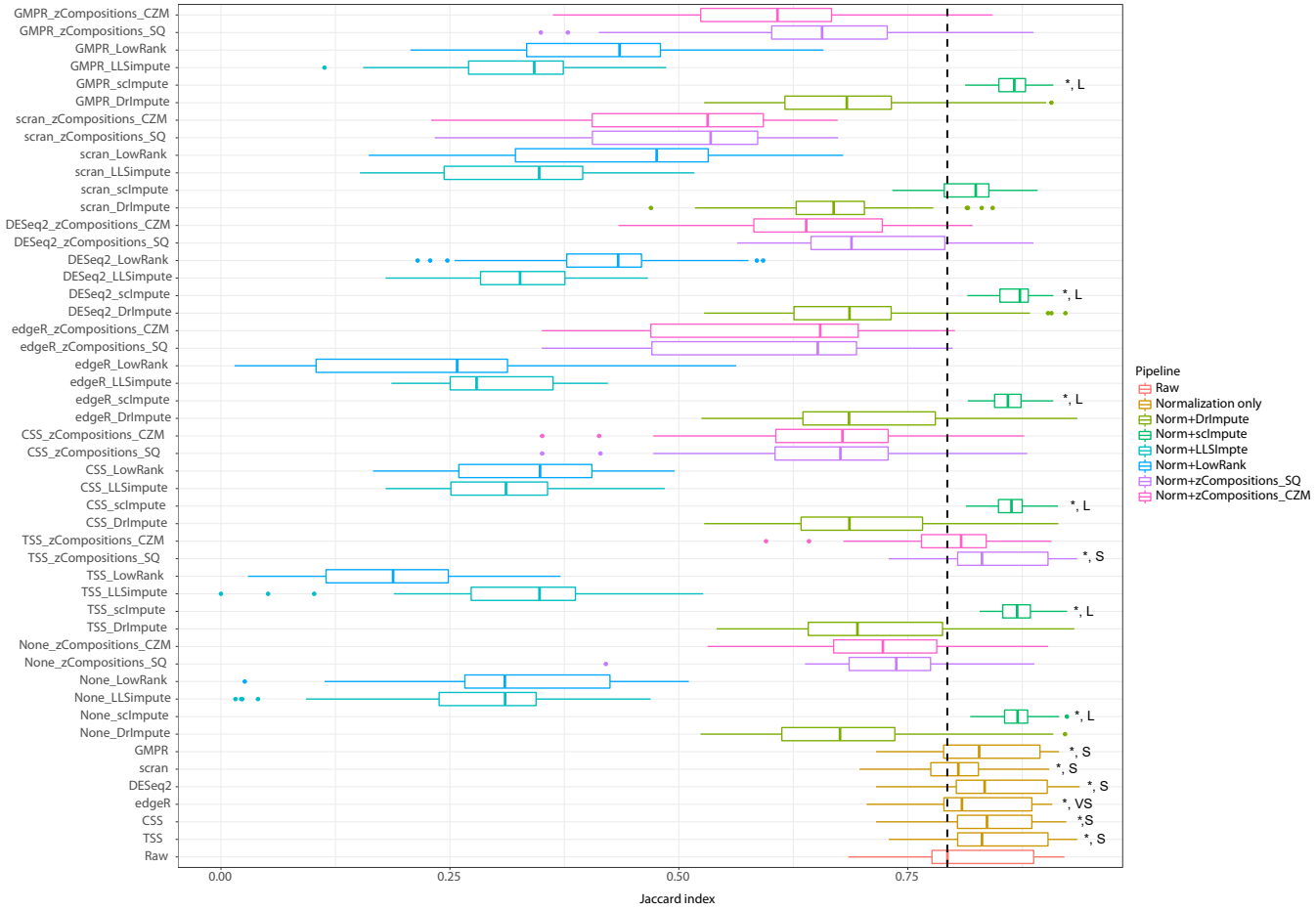


Fig. 8.16 Simulated Dataset 2 - Horizontal box plots of Jaccard index results obtained from comparisons between real and raw/pre-processed data. The median of values on raw data is highlighted by a dashed vertical line. The two labels separated by a comma on the right of each box plot indicate if significance was found in relation to raw data and the magnitude of the size effect, respectively.

Fig. 8.17 Simulated Dataset 3 - Horizontal box plots of Jaccard index results obtained from comparisons between real and raw/pre-processed data. The median of values on raw data is highlighted by a dashed vertical line. The two labels separated by a comma on the right of each box plot indicate if significance was found in relation to raw data and the magnitude of the size effect, respectively.

In this Chapter, a wide assessment on the influence of the choice of the pre-processing pipeline for 16S rDNA sequencing count data has been performed. This work was performed following the downstream analyses typically present in microbiome studies based on 16S rDNA-Seq data, studying the impact of the chosen pre-processing pipeline in terms of consequent changes in conclusions from the ones obtained using the ground truth data. More precisely, differences in sparsity, samples proportional abundance distributions, alpha and beta diversity indices and differential analyses were assessed.

Summarizing, *scImpute* pipelines turned out to be the overall best choice for 16S rDNA-Seq data preprocessing. In fact, despite some slight oscillation in performance between Simulations, it achieved optimal results in terms of recovered rare species and differential abundances detection and very satisfying performance in alpha and beta comparisons. It is noteworthy that in some cases (see Appendix B) normalizing data before imputing zero values slightly improved its performance, even though raw data are required in input by the tool specification. *DrImpute* showed optimal results in Simulation 1 for all the metrics, but it seemed more sensitive than *scImpute* to data characteristics, obtaining good but variable performance on the other Simulated Datasets. Regarding *zCompositions*, some differences in performance were obtained when considering its two modalities (SQ and CZM). In fact, even though they both left no zero values before imputation procedure, thus giving SMAPE and richness poor performance, the more refined SQ zero-imputation gave better results in terms of overall abundance profile reconstruction and differential analysis. However, results on abundance-based beta diversity were no acceptable for none of them. Finally, the overall worst performance was obtained by *LowRank* and *LLSimpute*, that obtained poor results in almost all the considered metrics. An overall qualitative summary of performance is reported in Table 8.15.

Table 8.15 Qualitative summary of overall pipelines performance. Increasing performance is denoted by zero to three stars, with corresponding colors from orange to dark green. Results are aggregated according to tested imputation methods.

	Sparsity	Proportional abundance	Richness	Pielou	Shannon	iSimpson	Tail	Beta (abundances)	Beta (p/a)	Differential analysis
scImpute	***	***	***	***	**	**	***	***	***	***
DrImpute	**	**	**	**	***	***	***	**	**	**
LLSimpute										
LowRank				*	*	*				
zCompositions_SQ		*		*	***	***	**			*
zCompositions_CZM		*		*	***	***				

Chapter 9

Conclusions

Next generation sequencing (NGS) has now become the most widely used approach to perform microbial community studies. In particular, the discovery of the 16S ribosomial RNA universal marker gene and the ever-decreasing experimental costs needed to sequence it made it the most chosen method for taxonomic studies. Among all NGS methodologies, targeted amplicon sequencing of the 16S ribosomial RNA (16S rRNA) gene is currently one of the most used strategies for the delineation and quantification of microbial population residing in a specific ecological niche. However, the appropriate treatment of the resulting data is still a very challenging issue, because of their characteristic extreme sparsity and variability. The main result of this thesis was the assessment of pre-processing of 16S rDNA-Seq count matrices approaches and the identification of optimal 16S rDNA-Seq pipelines. In particular:

- the state-of-the-art 16S rDNA-Seq pre-processing methods were identified and zero-imputation approaches available in literature that could be integrated in 16S rDNA-Seq work-flow were selected;
- a specific simulator to directly obtain realistic synthetic 16S rDNA-Seq count matrices was implemented;
- an extensive benchmark of pre-processing pipelines was conducted to obtain their performance assessment.

Identification of pre-processing 16S rDNA-Seq approaches.

The selection procedure of state-of-the-art 16S rDNA-Seq pre-processing methods led to the identification of little or no specific normalization techniques thought to work on metataxonomic studies data. The only two "tailored" tools for metagenomic studies were *GMPR*, a very recent tool that performs robust normalization for zero-inflated count data with

application to microbiome sequencing data, and CSS normalization, a method implemented in *metagenomeSeq* R package to account for biases specific to high-throughput sequencing microbial marker-gene survey data. In the majority of 16S rDNA-Seq data analyses, in fact, two of the most relevant tools for RNA-Seq data normalization are still used, i.e. *edgeR* and *DESeq2*, as well as the classical TSS normalization, that is transformation of the count matrix into a proportional abundance matrix dividing by the total number of reads per sample. Due to the similarity of metataxonimic data with scRNA-Seq data, also a new normalization method introduced for this field, *scran*, was considered because peculiarly addressing normalization in case of sparse count data.

Regarding zero-imputation step, no tools were available at the best of our knowledge that were specifically designed for this framework. However, two recently published tools that dealt with compositional zero count recovery, *robCompositions* and *zCompositions*, were found and thus included in this work. In addition, two of the most-promising scRNA-Seq zero-imputation tools , *DrImpute* and *scImpute*, were considered because no assumption strictly related to scRNA framework was done by these tool for zero values imputation, thus permitting a direct application also on 16S rDNA-Seq data. Finally, two other zero-imputation methods from other fields, *Low-Rank* and *LLSimpute*, were included for the same reason in addition to their original implementation motivation. In fact, *Low-Rank* is a missing values recovery method made to be used in situation where sparsity reaches critical levels, such as in the case of the Netflix problem, while *LLSimpute* was proposed for sparse microarray gene expression data.

Implemetation of 16S rDNA-Seq count data simulator.

To the best of our knowledge, the majority of available metagenomic simulators are intended for synthetic read production. This implies they are very useful tools for testing methods for data treatment prior to OTU table formation, but they are not useful to test count data pre-processing and analysis steps. Additionally, all the very few approaches available in literature for synthetic count data simulation in literature use the Negative Binomial model, a modellization that was found to be very appropriate for bulk RNA-Seq data but that does not fit extreme sparsity levels present in 16S microbiome data matrices. The simulator implemented in the context of this thesis, metaSPARSim, allows for simple and rapid production of 16S count data with proven similar features to real datasets. As no standard procedures are available for count simulators performances, we decided to evaluate metaSPARSim in its ability of reproducing real data characteristics, encoded in three main features: sparsity, intensity and variability. Tests for performance assessment were performed using a pool of three real datasets with very different intrinsic characteristics, to evaluate the goodness of metaSPARSim simulations in different scenarios. Additionally, the generative

power of our tool was investigated considering two different subsampling size from HMP data, one composed by 5 replicates per group and the other composed by 100 replicates.

In the performed tests, metaSPARSim demonstrated to be able to generate realistic count data in all the considered scenarios. In fact, sparsity, intensity and variability distributions of simulated datasets were well reproduced for all the three datasets. The raw milk cheese dataset offered a very challenging framework, due to the low number of biological replicates per group and the extreme sparsity level (97%). These dataset characteristics necessarily imply the estimate of parameters to be less robust than in other datasets. On the contrary, the third dataset, derived from HMP data, had no difficulties linked to the number of replicates per group, by rather with their nature. In fact, in that case replicates came from different subjects and this added interindividual variability to the natural biologic one. However, tests on both little and large population taken from HMP data and on raw milk cheese data revealed the robustness of the simulator to these challenging situations, showing its ability in capturing and reproducing the main characteristics of data and returning simulated count matrices that greatly resembled the real ones.

The availability of count data simulators is extremely valuable for methods developers, which can exploit the ground truth provided by simulated data to test and validate their tools. In addition, simulated data could be useful even for end users who want to find the most accurate analysis methods among the many available in literature to use in their work. Indeed, the availability of simulated data allows to assess state of the art analysis tools and so to identify the more suitable one for the specific scenario. Thus, we believe that metaSPARSim could be a valuable tool also for other researchers involved in developing, testing and using robust and reliable data analysis methods in the context of 16S rDNA-Seq and scRNA-Seq.

Benchmark of pre-processing pipelines.

The 16S rDNA-Seq approaches identified during the first task were used to compose different pre-processing pipelines. In particular, in the benchmark were included both 6 only-normalization pipelines, that reflected currently adopted 16S data treatment work-flow, and pipelines with zero-imputation step. Of these, 6 were composed by the only zero-imputation step, while the remaining 36 were made by the combination of a normalization step plus the zero-imputation one. Therefore, a total of 48 pre-processing pipeline were evaluated using three very dissimilar metaSPARSim-generated datasets, differing for experimental characteristics (sequencing depth, number of samples and replicate and number of subgroups, DNA/RNA starting material), for replicates nature (biological replicates versus different subjects sampling) and for sparsity level (78-97%). It is noteworthy that different sequencing depths imply different partitioning of sparsity in biological and sequencing-derived. In fact,

a high sequencing depth allows for a very contained loss of rare species, thus implying that observed zeros are more likely to be true biological zeros.

The 48 pipelines were evaluated using quantitative measures (SMAPE, Aitchison's distance) to assess differences on proportional abundances between the ground truth, the raw and the pre-processed datasets. Additionally, the influence of different pre-processing choices was evaluated looking at repercussions on typical microbiome studies analyses. In particular, effects on alpha and beta diversity, subpopulation profiling and differential abundance testing were evaluated and compared.

Similarly to what obtained for scRNA-Seq data, the introduction of zero-imputation step using methods designed for sparse sequencing data permitted to recover the lost information very well and not to introduce unwanted biases. In our tests, the normalization step showed indeed an improvement of zero-imputation performances compared to zero-imputation-only pipelines, also for tools that explicitly ask for non-normalized counts, such as *scImpute*. However, the improvement was of similar entity when applying different normalization tools, with a slight higher performance for *GMPR* tool.

Despite being very interesting and well-structured imputation strategies, the solutions proposed in *robCompositions* had to be excluded from the benchmark. In fact, the knn imputation implementation showed not to manage situations in which too many features with null values are present in the analyzed dataset. This caused the impossibility of using knn imputed values to initialize the second implemented solution, the iterative algorithm, that was consequently also excluded. For this last, also a solution with 0.001 pseudocount addition to initialize the iterative procedure was tried with no solution, because the high percentage of null values still represented a problem for regression methods, that all stopped or had to be stopped before hours of running time with no solution reached.

The other compositional tool, *zCompositions*, was tested with two different configurations, named "CZM" and "SQ". The pipelines including this tool resulted to be very good in preserving overall proportional distribution, but made poor selection on null values between the ones in need of imputation and the ones that reflected true species absence. As a consequence, their performance in richness recovery, i.e. in stating really present species, were very poor in all tested datasets.

Low-Rank and *LLSimpute* were found to be not adequate for 16S rDNA-Seq data pre-processing in all tested datasets and according to all the selected metrics, thus demonstrating the non-transferability of these techniques into 16S microbiome studies framework.

The most performing, reliable and robust pipelines were the ones that included *scImpute* and *DrImpute* tools, combined with whichever normalization method. These pipelines showed very good performances both in retrieving truly present species an in reconstructing

proportional abundance levels. They both had a little drop in performances in some sporadic cases. In fact, *scImpute* tended to generate some isolated error of appreciable entity when applied to the first dataset, while *drImpute* showed some difficulties in identifying true zeros in the third dataset, where sequencing zeros were reduced to a minimal part. However, these two tools demonstrated good overall performances in all tested dataset and a great improvement in downstream analysis correctness in relation to raw or simply-normalized data.

Benchmark results are, consequently, important for two main reasons: (i) they permitted to find the best performing pipelines that could be then solidly chosen for data pre-processing in future real data analyses and (ii) they showed how the introduction of a well-performed zero-imputation step in 16S rDNA sequencing data would produce more correct and reliable analysis results.

Final remarks.

Following the results obtained in this work, real data analyses on the two experiments performed in the context of this thesis are currently being performed. The identification of the best pipelines for 16S rDNA sequencing pre-processing allows now for a robust choice of tools to perform this step that avoid misleading results in downstream analyses. Future analyses on the temporal series derived from both the proprietary datasets include also microbial networks reconstruction, for the accurate derivation of which the obtained best practices are of pivotal importance. In fact, their use will guarantee more robust and reliable data, permitting a more solid identification of relations internal to the microbiomes.

The current trend of decreasing in costs for 16S rDNA sequencing could lead to the future possibility of performing studies in which demultiplexing level could be heavily decreased and, consequently, the loss of information due to sequencing could also be reduced. Nevertheless, this advantage would be more probably used by researchers to perform a higher number of replicates to obtain more robust results, in spite of sequencing less samples with a higher sequencing depth. This would imply, again, the need for a good treatment of zero values, in order to reconstruct the portion of rare species that were unobserved.

The great importance of treating 16S rDNA sequencing data as compositional suggests that room for improvement could also be found in the metrics adopted to quantify microbiomes diversity. In fact, more appropriate measures of alpha and beta diversity could be introduced to include this particular and fundamental aspect of these data.

The ever-increasing use of 16S rDNA sequencing approach to microbiome studies and the increasing will to adopt this technique in clinical routines impose reliable tools to treat the resulting data and experimental and computational standards to be defined. To this aim, a

lot of work has yet to be done to fully understand and describe metataxonomics data, that nowadays still constitute a great challenge.

Appendix A

metaSPARSim performance assessment: HMP with large sample size

Here are reported the details on results of the performance assessment using the HMP-derived dataset with large sample size.

Sparsity

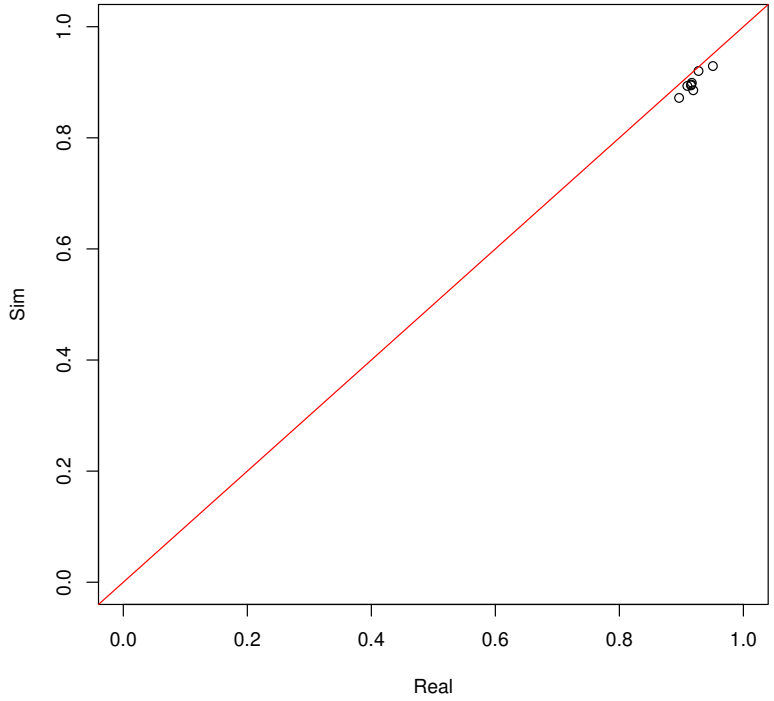
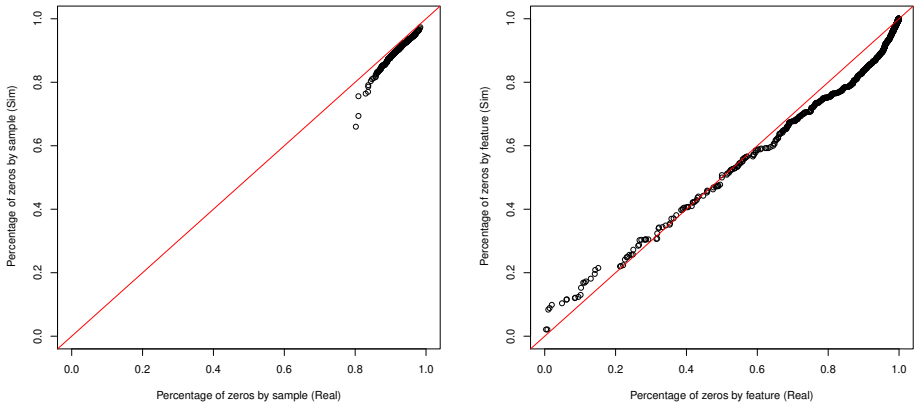


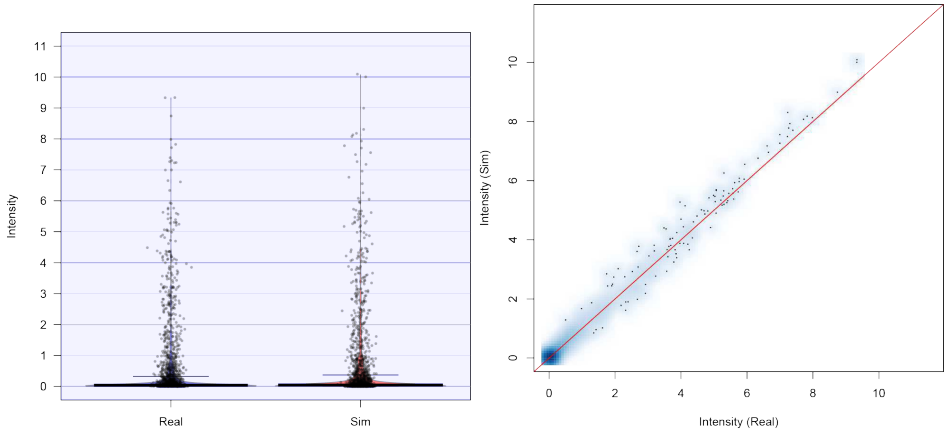
Fig. A.1 Q−Q plot of group-specific percentage of zeros in real and simulated datasets.



(A) (B)

Fig. A.2 Q−Q plot of percentage of zeros in real and simulated datasets, calulated by sample (A) and by feature (B).

Intensity



(A) (B)

Fig. A.3 Comparison of Log2 count intensity in real and simulated datasets, represented as RDI plot (A) and scatter plot (B).

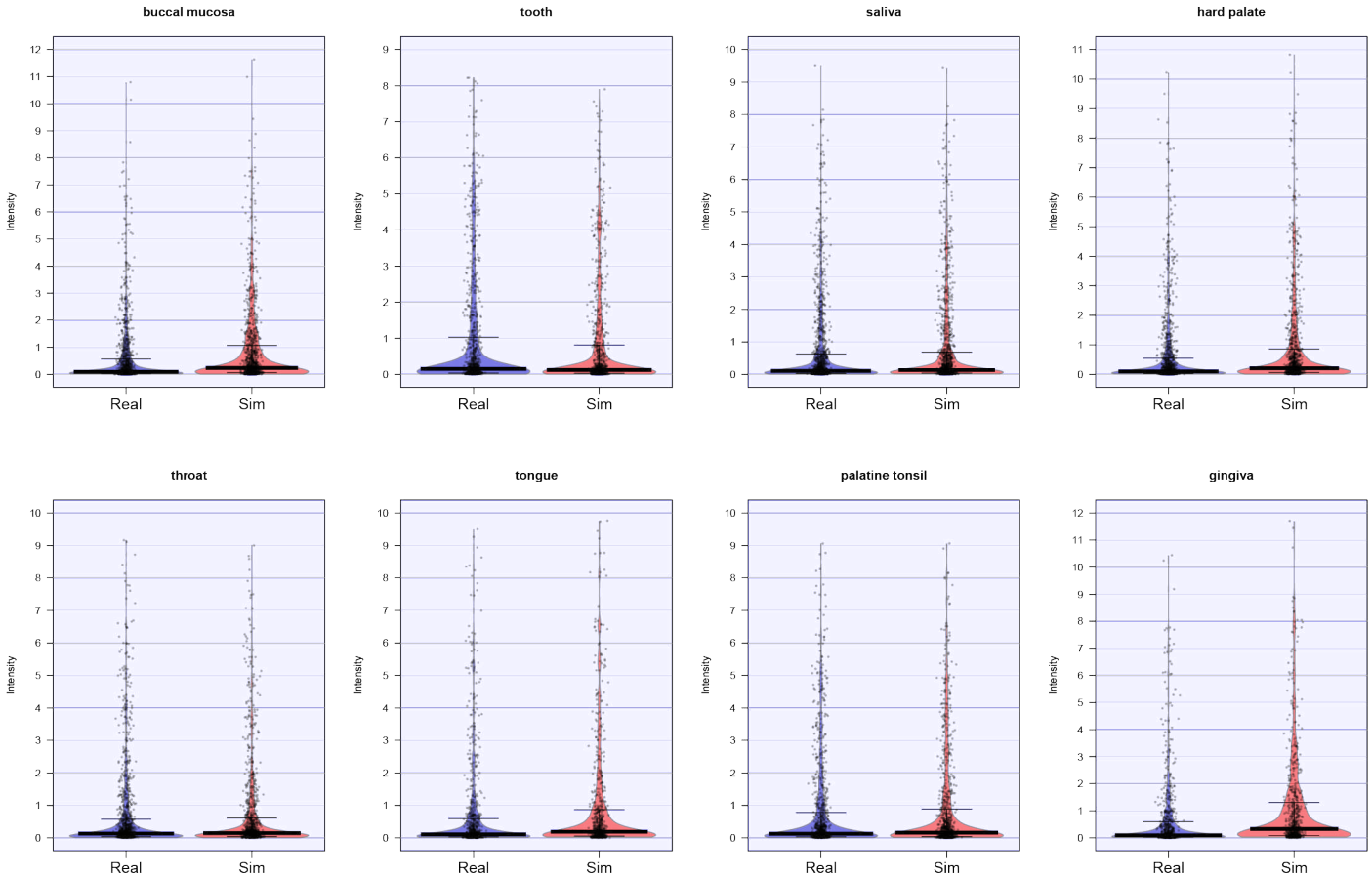


Fig. A.4 RDI plots of Log2 count values of real and simulated data within each group, excluding null counts.

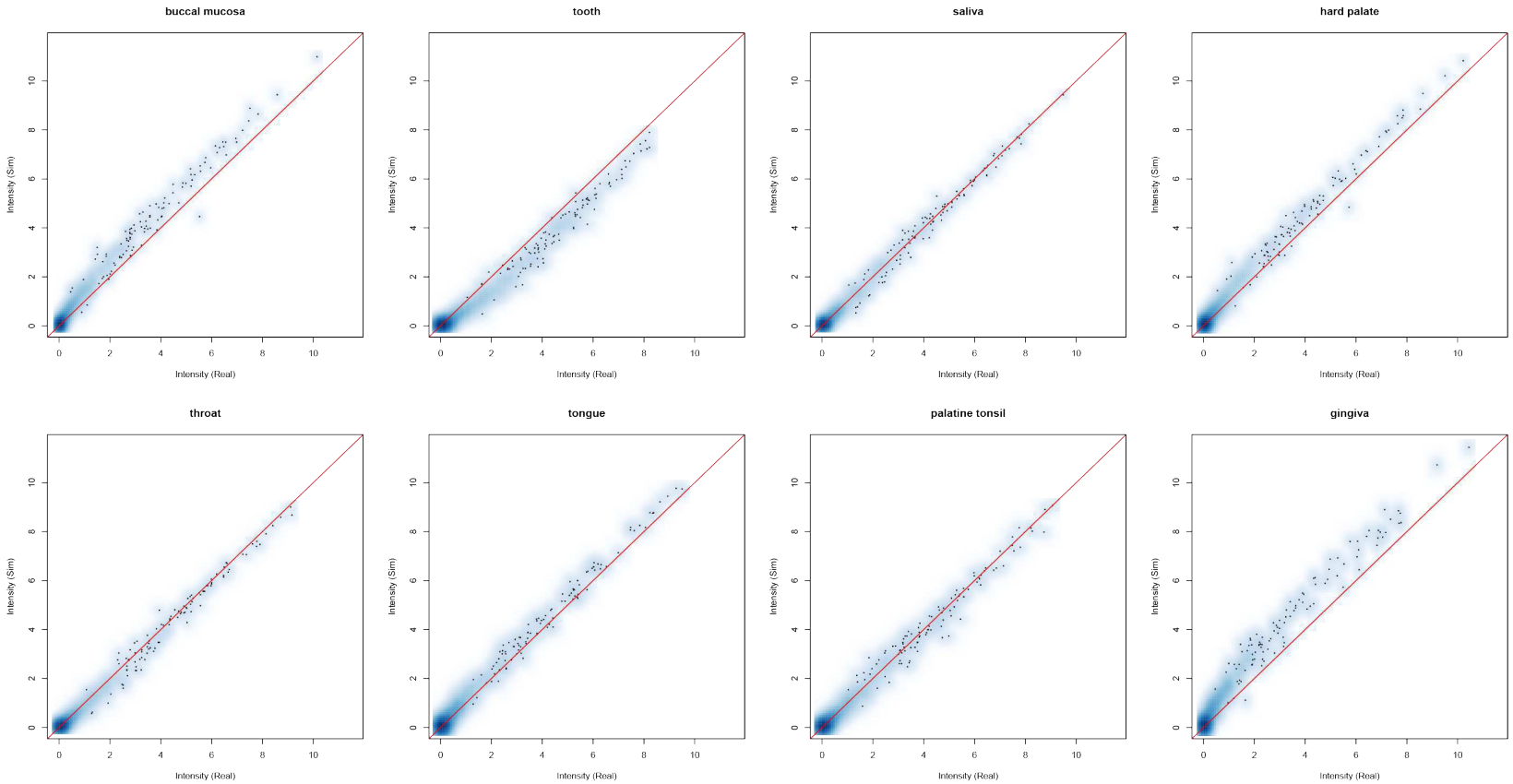


Fig. A.5 Scatter plots of Log2 count values of real and simulated data within each group, excluding cases with zero counts in both data.

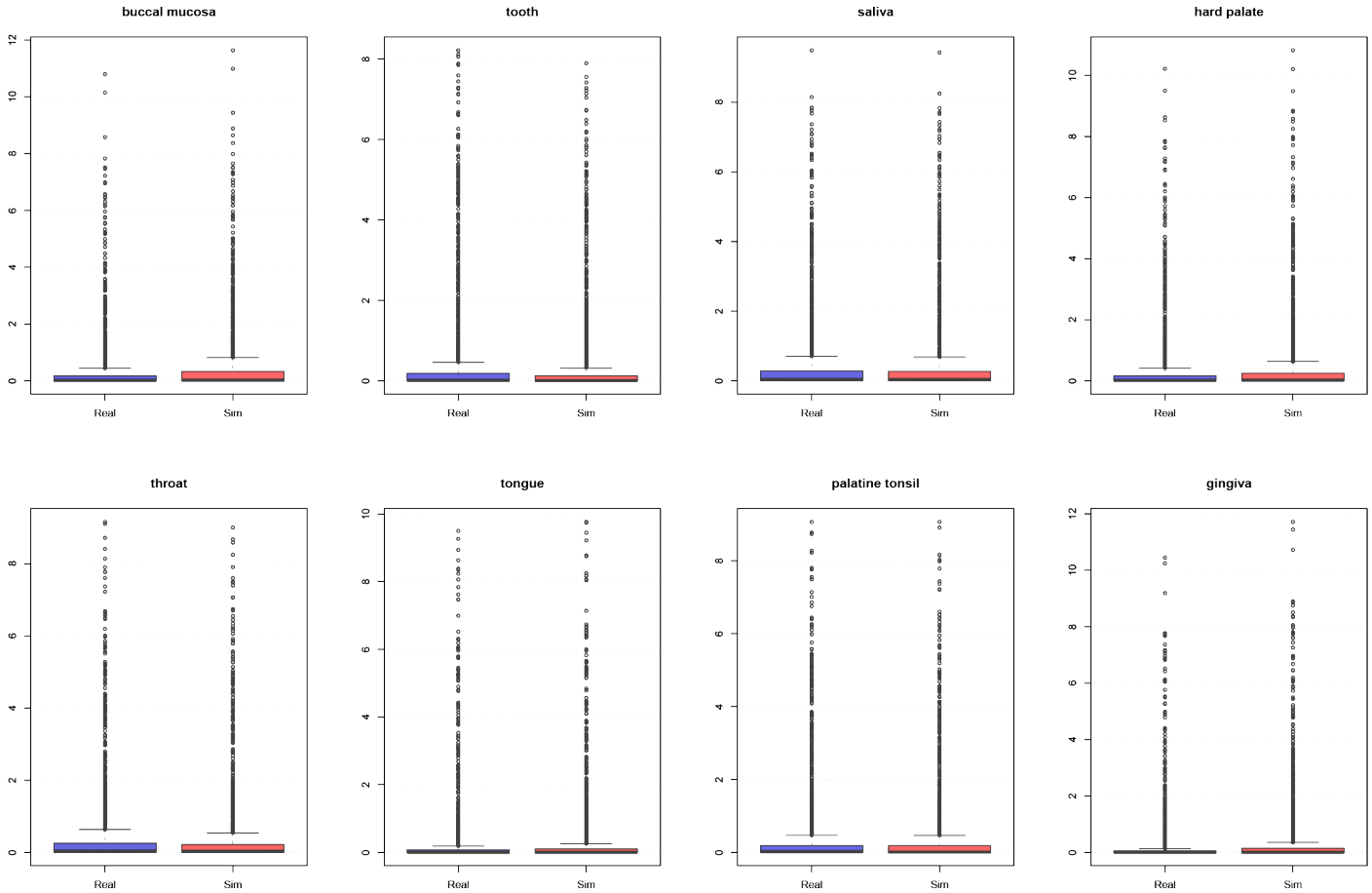


Fig. A.6 Box plots of Log2 count values of real and simulated data within each group.

Table A.1 Mann-Whitney U test and effect size results in comparing real and simulated mean count distribution within groups.

Group	*P* value	Significance	Cohen's *d* magnitude	Bootstrap significance (%)
1	0.291	N	– – –	0
2	0.002	Y	Negligible	0
3	0.145	N	– – –	0
4	0.647	N	– – –	0
5	0.028	Y	Negligible	0
6	0.255	N	– – –	0
7	0.143	N	– – –	0
8	0.65	N	– – –	0

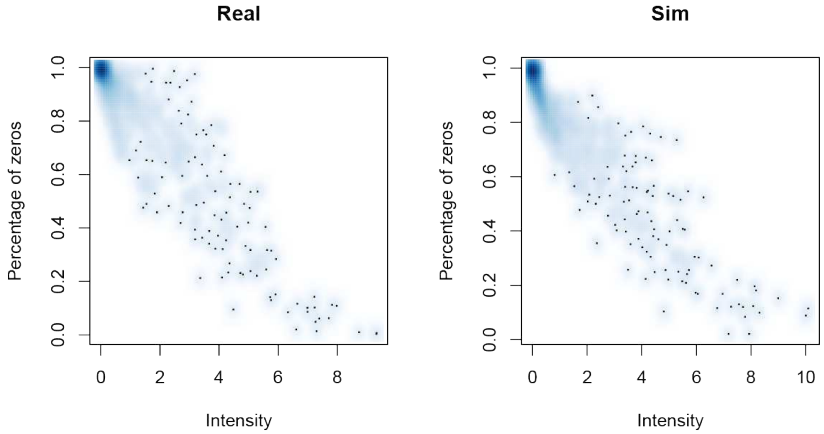
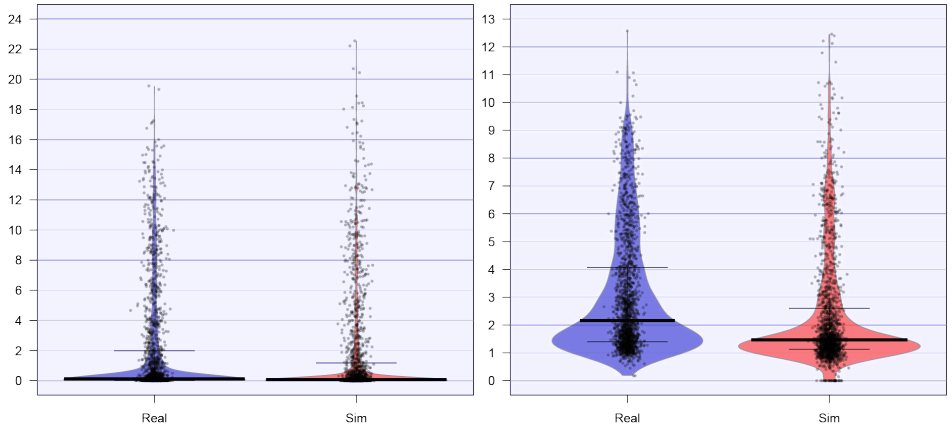


Fig. A.7 Scatter plot of the overall relation between Log2 count intensity and sparsity in real and simulated datasets.

Variability



(A) (B)

Fig. A.8 Comparison of Log2 variability values in real and simulated datasets, calculated as variance (A) and RV (B).

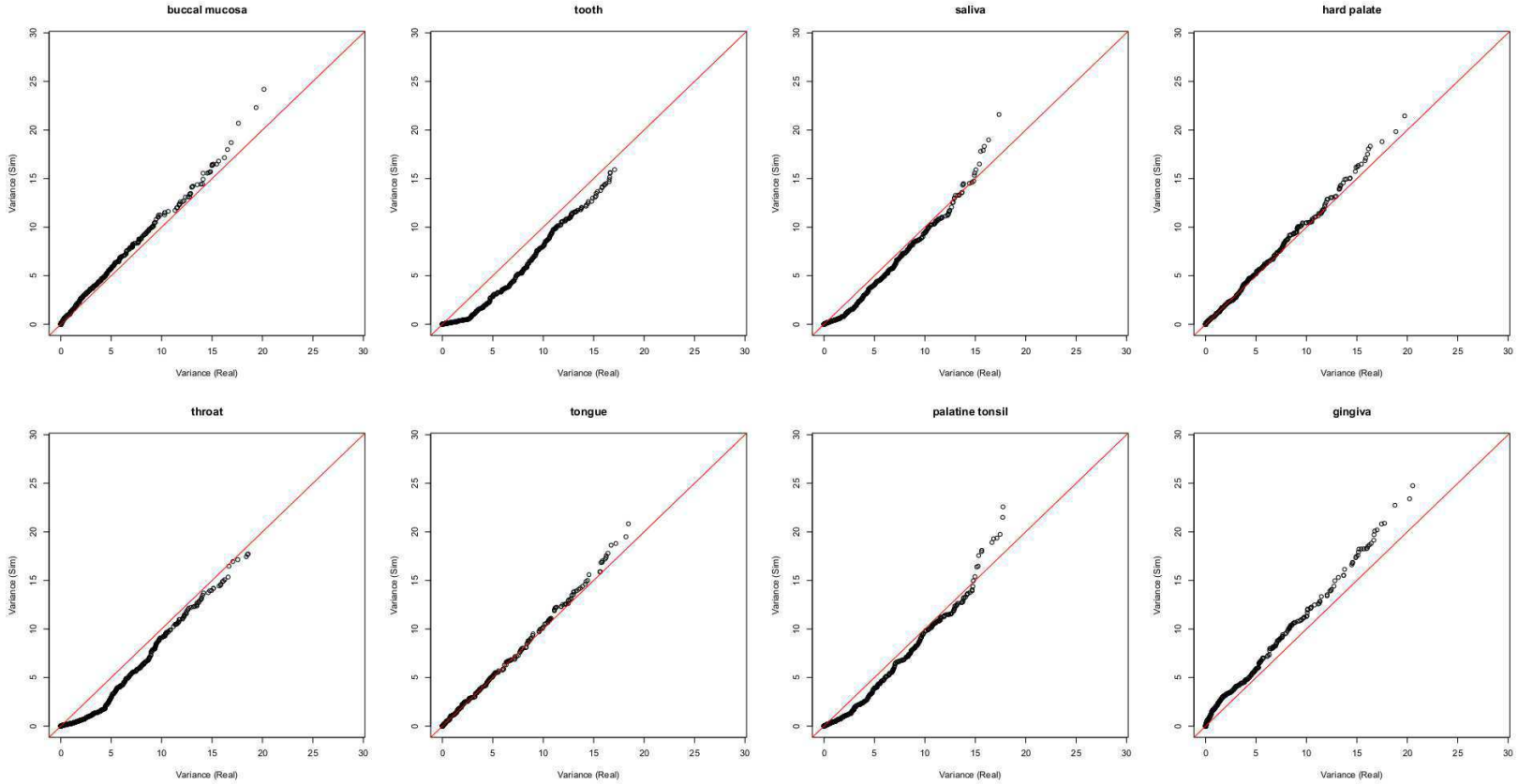


Fig. A.9 Q−Q plots of of Log2 variance values in real and simulated datasets within each group.

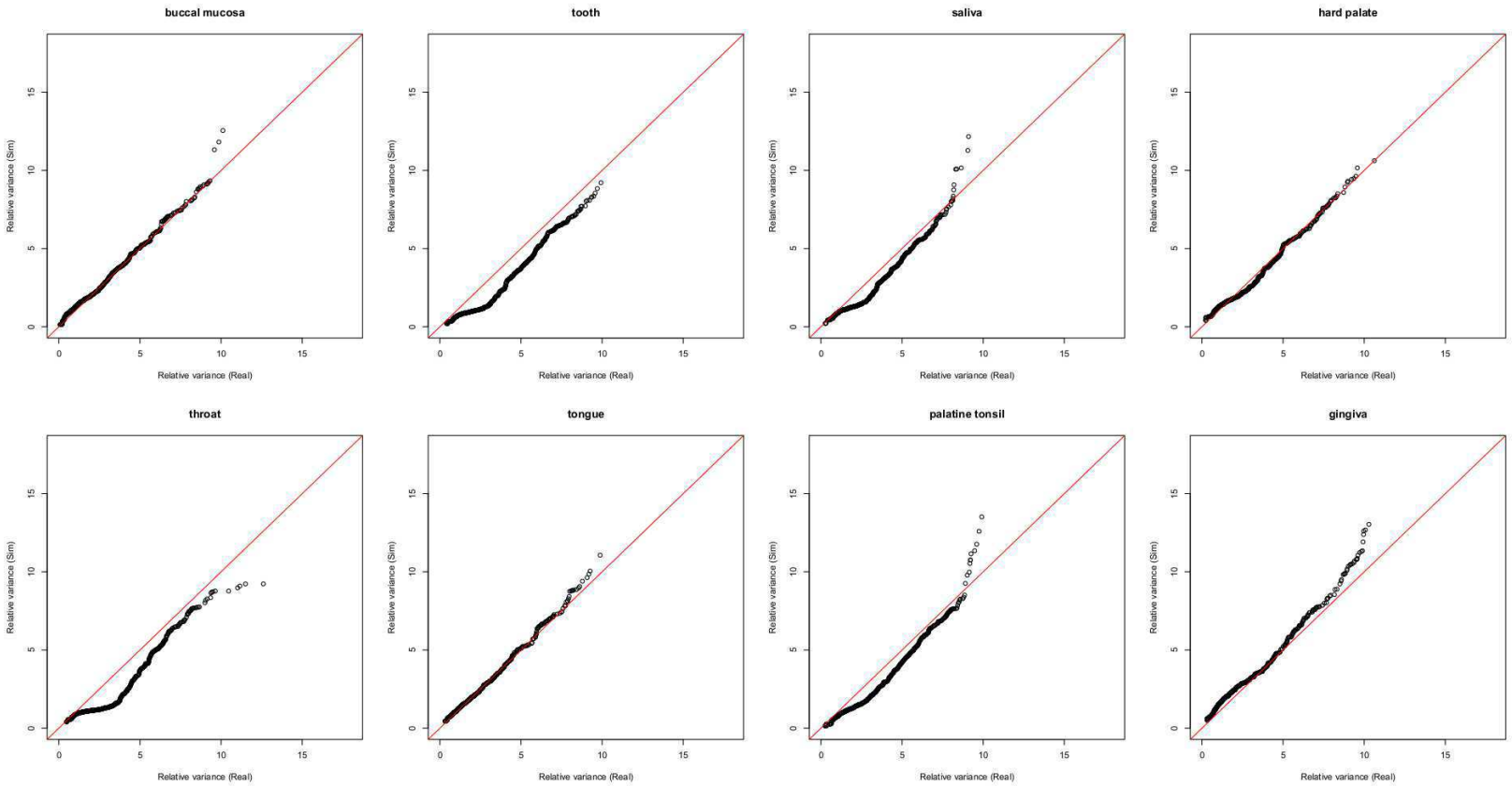


Fig. A.10 Q−Q plots of of Log2 RV values in real and simulated datasets within each group.

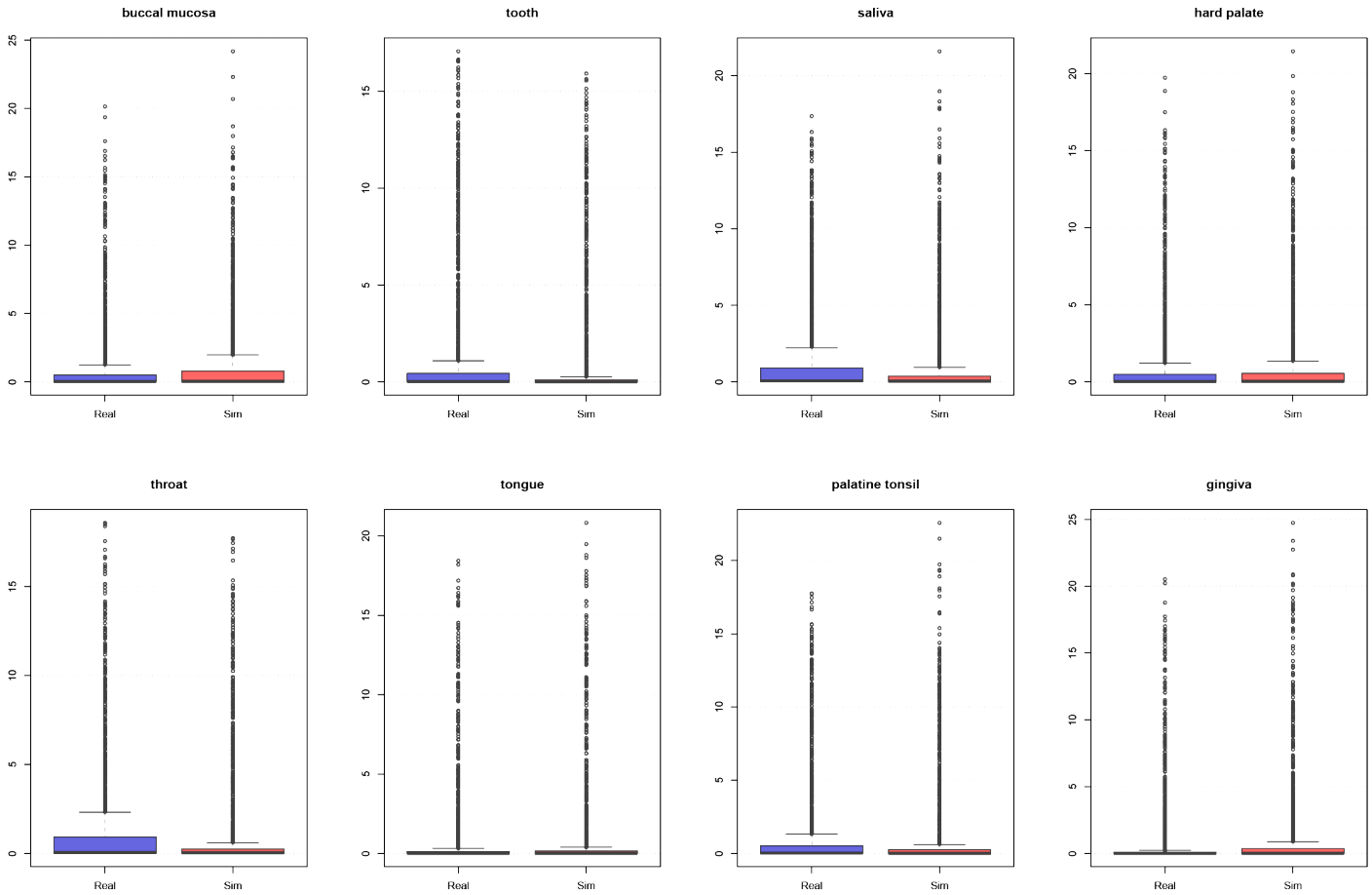


Fig. A.11 Box plots of of Log2 variance values in real and simulated datasets within each group.

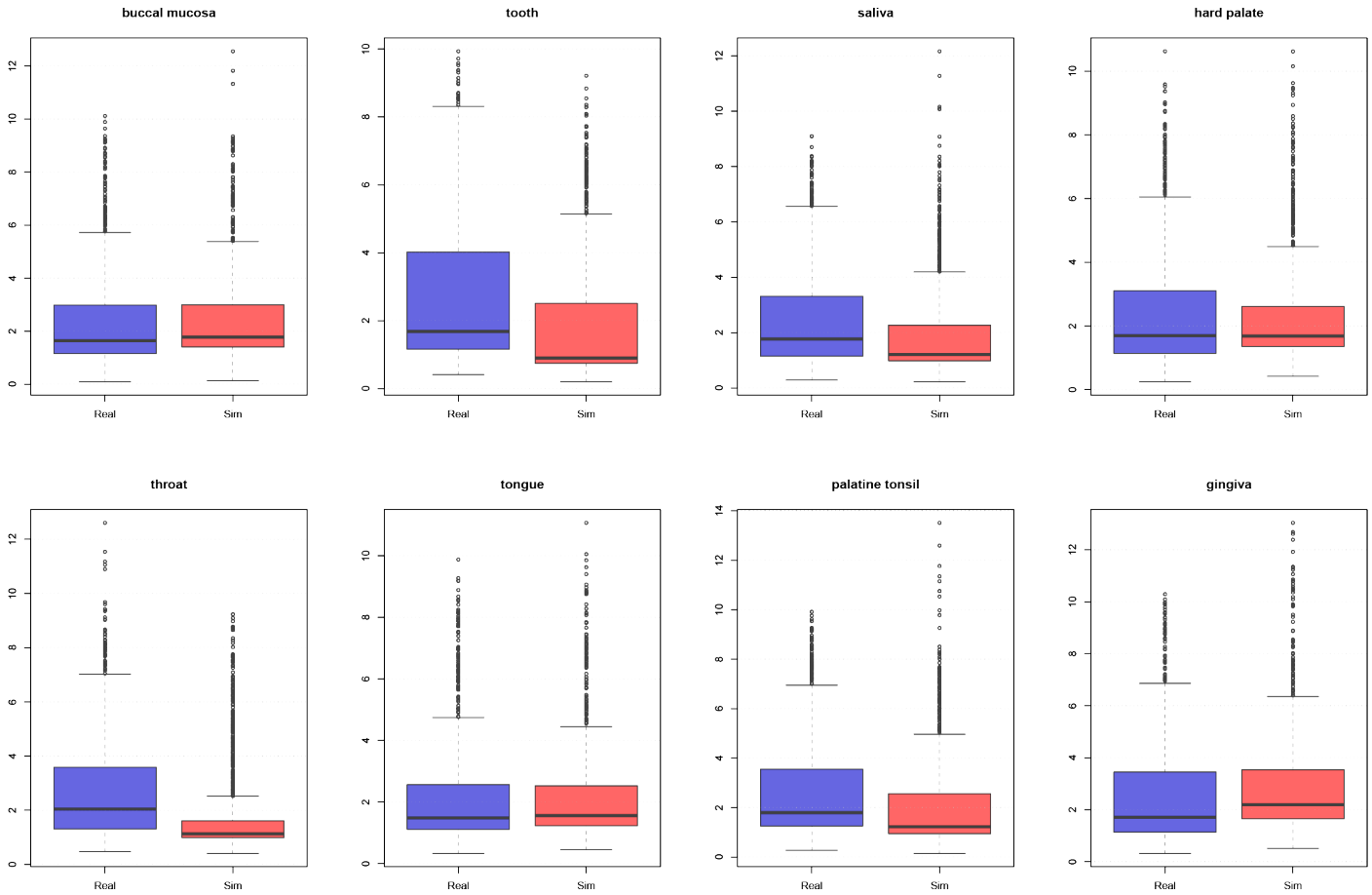


Fig. A.12 Box plots of of Log2 RV values in real and simulated datasets within each group.

Table A.2 Mann-Whitney U test and effect size results in comparing real and simulated mean count distribution within groups.

Group	*P* value	Significance	Cohen's *d* magnitude	Bootstrap significance (%)
1	0.73	N	– – –	0
2	0	Y	Negligible	0.12
3	0	Y	Negligible	0.01
4	0.647	N	– – –	0
5	0	Y	Negligible	0.86
6	0.145	N	– – –	0
7	0.02	Y	Negligible	0
8	0.845	N	– – –	0

Appendix B

Results of the benchmark of pre-processing pipelines on 16S count data: details.

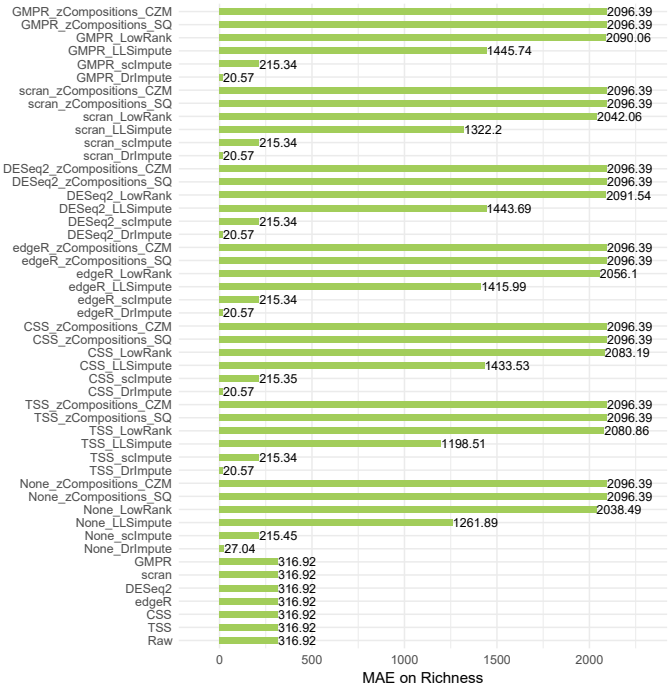


Fig. B.1 Simulated Dataset 1 - MAE between richness index calculated on real and on raw and pre-processed data.

Table B.1 Simulated Dataset 1, count matrix sparsity. Real, raw and pre-processed datasets are ordered for decreasing similarity with true sparsity. In the "Behaviour" column, datasets with overestimated sparsity are labelled with "O", while underestimation is labelled with "U".

Pipeline	Sparsity	Behaviour
Real	63.03 %	O
None_DrImpute	62.74 %	U
TSS_DrImpute	62.65 %	U
CSS_DrImpute	62.65 %	U
edgeR_DrImpute	62.65 %	U
DESeq2_DrImpute	62.65 %	U
scran_DrImpute	62.65 %	U
GMPR_DrImpute	62.65 %	U
TSS_scImpute	69.5 %	O
edgeR_scImpute	69.5 %	O
DESeq2_scImpute	69.5 %	O
scran_scImpute	69.5 %	O
GMPR_scImpute	69.5 %	O
CSS_scImpute	69.51 %	O
None_scImpute	69.51 %	O
After_sequencing	72.56 %	O
TSS	72.56 %	O
CSS	72.56 %	O
edgeR	72.56 %	O
DESeq2	72.56 %	O
scran	72.56 %	O
GMPR	72.56 %	O
TSS_LLSimpute	27 %	U
None_LLSimpute	25.09 %	U
scran_LLSimpute	23.28 %	U
edgeR_LLSimpute	20.46 %	U
CSS_LLSimpute	19.93 %	U
DESeq2_LLSimpute	19.62 %	U
GMPR_LLSimpute	19.56 %	U
None_LowRank	1.74 %	U
scran_LowRank	1.65 %	U
edgeR_LowRank	1.21 %	U
TSS_LowRank	0.47 %	U
CSS_LowRank	0.4 %	U
GMPR_LowRank	0.19 %	U
DESeq2_LowRank	0.15 %	U
None_zCompositions_SQ	0 %	U
None_zCompositions_CZM	0 %	U
TSS_zCompositions_SQ	0 %	U
TSS_zCompositions_CZM	0 %	U
CSS_zCompositions_SQ	0 %	U
CSS_zCompositions_CZM	0 %	U
edgeR_zCompositions_SQ	0 %	U
edgeR_zCompositions_CZM	0 %	U
DESeq2_zCompositions_SQ	0 %	U
DESeq2_zCompositions_CZM	0 %	U
scran_zCompositions_SQ	0 %	U
scran_zCompositions_CZM	0 %	U
GMPR_zCompositions_SQ	0 %	U
GMPR_zCompositions_CZM	0 %	U

Table B.2 Simulated Dataset 2, count matrix sparsity. Real, raw and pre-processed datasets are ordered for decreasing similarity with true sparsity. In the "Behaviour" column, datasets with overestimated sparsity are labelled with "O", while underestimation is labelled with "U".

Pipeline	Sparsity	Behaviour
Real	56.61 %	O
None_scImpute	55.85 %	U
GMPR_scImpute	55.84 %	U
DESeq2_scImpute	55.84 %	U
scran_scImpute	55.84 %	U
TSS_scImpute	55.84 %	U
CSS_scImpute	55.84 %	U
edgeR_scImpute	55.84 %	U
Raw	67.91 %	O
TSS	67.91 %	O
CSS	67.91 %	O
edgeR	67.91 %	O
DESeq2	67.91 %	O
scran	67.91 %	O
GMPR	67.91 %	O
None_DrImpute	42.32 %	U
TSS_DrImpute	42.32 %	U
CSS_DrImpute	42.32 %	U
edgeR_DrImpute	42.32 %	U
DESeq2_DrImpute	42.32 %	U
scran_DrImpute	42.32 %	U
GMPR_DrImpute	42.32 %	U
edgeR_LLSimpute	24.99 %	U
DESeq2_LLSimpute	24.19 %	U
CSS_LLSimpute	24.05 %	U
scran_LLSimpute	24.02 %	U
None_LLSimpute	23.42 %	U
TSS_LLSimpute	22.42 %	U
GMPR_LLSimpute	20.08 %	U
scran_LowRank	5.08 %	U
TSS_LowRank	1.38 %	U
edgeR_LowRank	1.34 %	U
CSS_LowRank	0.56 %	U
None_LowRank	0 %	U
None_zCompositions_SQ	0 %	U
None_zCompositions_CZM	0 %	U
TSS_zCompositions_SQ	0 %	U
TSS_zCompositions_CZM	0 %	U
CSS_zCompositions_SQ	0 %	U
CSS_zCompositions_CZM	0 %	U
edgeR_zCompositions_SQ	0 %	U
edgeR_zCompositions_CZM	0 %	U
DESeq2_LowRank	0 %	U
DESeq2_zCompositions_SQ	0 %	U
DESeq2_zCompositions_CZM	0 %	U
scran_zCompositions_SQ	0 %	U
scran_zCompositions_CZM	0 %	U
GMPR_LowRank	0 %	U
GMPR_zCompositions_SQ	0 %	U
GMPR_zCompositions_CZM	0 %	U

Table B.3 Simulated Dataset 3, count matrix sparsity. Real, raw and pre-processed datasets are ordered for decreasing similarity with true sparsity. In the "Behaviour" column, datasets with overestimated sparsity are labelled with "O", while underestimation is labelled with "U".

Pipeline	Sparsity	Behaviour
Real	91.26 %	O
After_sequencing	94.34 %	O
TSS	94.34 %	O
CSS	94.34 %	O
edgeR	94.34 %	O
DESeq2	94.34 %	O
scran	94.34 %	O
GMPR	94.34 %	O
GMPR_scImpute	87.08 %	U
DESeq2_scImpute	87.08 %	U
scran_scImpute	87.07 %	U
None_scImpute	87.07 %	U
CSS_scImpute	87.06 %	U
TSS_scImpute	87.06 %	U
edgeR_scImpute	87.05 %	U
TSS_DrImpute	81.86 %	U
edgeR_DrImpute	81.85 %	U
CSS_DrImpute	79.28 %	U
DESeq2_DrImpute	79.07 %	U
scran_DrImpute	79.07 %	U
GMPR_DrImpute	79.07 %	U
None_DrImpute	79.03 %	U
TSS_LLSimpute	25.57 %	U
None_LLSimpute	23.73 %	U
DESeq2_LLSimpute	23.56 %	U
CSS_LLSimpute	22.72 %	U
GMPR_LLSimpute	22.71 %	U
scran_LLSimpute	22.48 %	U
edgeR_LLSimpute	20.33 %	U
TSS_LowRank	9.53 %	U
edgeR_LowRank	7.31 %	U
CSS_LowRank	4.21 %	U
GMPR_LowRank	3.32 %	U
scran_LowRank	1.31 %	U
DESeq2_LowRank	1.08 %	U
None_LowRank	0.22 %	U
None_zCompositions_SQ	0 %	U
None_zCompositions_CZM	0 %	U
TSS_zCompositions_SQ	0 %	U
TSS_zCompositions_CZM	0 %	U
CSS_zCompositions_SQ	0 %	U
CSS_zCompositions_CZM	0 %	U
edgeR_zCompositions_SQ	0 %	U
edgeR_zCompositions_CZM	0 %	U
DESeq2_zCompositions_SQ	0 %	U
DESeq2_zCompositions_CZM	0 %	U
scran_zCompositions_SQ	0 %	U
scran_zCompositions_CZM	0 %	U
GMPR_zCompositions_SQ	0 %	U
GMPR_zCompositions_CZM	0 %	U

Table B.4 Simulated Dataset 1 - Pipelines median SMAPE and Aitchison's distance to the ground truth and their ranking.

Pipeline	SMAPE	Rank (SMAPE)	Aitchison dist	Rank (Aitchison dist)
GMPR_DrImpute	7.416015	1	18.740302	6
None_DrImpute	7.420569	2	18.580886	3
CSS_DrImpute	7.538333	3	18.729170	5
DESeq2_DrImpute	7.553571	4	18.664090	4
scran_DrImpute	7.768557	5	18.543075	2
edgeR_DrImpute	7.975936	6	18.436698	1
TSS_DrImpute	7.981157	7	18.944255	9
DESeq2_scImpute	11.19143	8	26.409517	35
CSS_scImpute	11.19223	9	26.407097	29
TSS_scImpute	11.19246	10	26.407717	32
edgeR_scImpute	11.19275	11	26.408625	34
GMPR_scImpute	11.19313	12	26.408523	33
None_scImpute	11.19364	13	26.407245	30
scran_scImpute	11.19373	14	26.407560	31
CSS	12.10931	15	19.039236	11
GMPR	12.10931	15	19.039236	11
Raw	12.10931	15	19.039236	11
DESeq2	12.10931	15	19.039236	11
edgeR	12.10931	15	19.039236	11
scran	12.10931	15	19.039236	11
TSS	12.10931	15	19.039236	11
CSS_zCompositions_CZM	72.84332	22	21.733091	25
DESeq2_zCompositions_CZM	72.90399	23	21.818842	26
GMPR_zCompositions_CZM	73.03085	24	22.253489	27
scran_zCompositions_CZM	73.13219	25	22.764224	28
None_zCompositions_CZM	73.27001	26	20.249661	24
TSS_LowRank	73.29568	27	75.693407	42
None_LowRank	73.30589	28	71.782142	41
edgeR_LowRank	73.86602	29	62.987521	40
TSS_zCompositions_CZM	74.03355	30	18.815973	8
CSS_LowRank	74.08447	31	61.471982	39
edgeR_LLSimpute	74.16727	32	93.236953	43
edgeR_zCompositions_CZM	74.47802	33	18.815248	7
DESeq2_LowRank	74.61729	34	60.028589	38
GMPR_LowRank	74.78925	35	57.331938	37
None_LLSimpute	74.89395	36	100.102560	44
TSS_LLSimpute	75.30066	37	118.484365	48
scran_LowRank	75.46973	38	35.595308	36
None_zCompositions_SQ	77.2296	39	19.170769	22
GMPR_zCompositions_SQ	77.25703	40	19.185817	23
DESeq2_zCompositions_SQ	77.25954	41	19.164941	21
CSS_zCompositions_SQ	77.26329	42	19.137127	20
TSS_zCompositions_SQ	77.33451	43	19.107168	19
edgeR_zCompositions_SQ	77.33693	44	19.080997	18
scran_zCompositions_SQ	77.41161	45	19.026737	10
scran_LLSimpute	78.34555	46	121.003314	49
DESeq2_LLSimpute	80.23056	47	105.101429	45
GMPR_LLSimpute	80.50812	48	107.936310	47
CSS_LLSimpute	81.20288	49	107.500500	46

Table B.5 Simulated Dataset 2 - Pipelines median SMAPE and Aitchison's distance to the ground truth and their ranking.

Pipeline	SMAPE	Rank (SMAPE)	Aitchison dist	Rank (Aitchison dist)
DESeq2_scImpute	12.049579	1	25.111792	21
GMPR_scImpute	12.055424	2	25.110521	20
scran_scImpute	12.056787	3	25.113207	26
CSS_scImpute	12.058242	4	25.112180	23
edgeR_scImpute	12.060855	5	25.112818	25
TSS_scImpute	12.061035	6	25.112770	24
None_scImpute	12.070743	7	25.111918	22
Raw	15.493293	8	23.753575	13
CSS	15.493293	8	23.753575	13
DESeq2	15.493293	8	23.753575	13
edgeR	15.493293	8	23.753575	13
GMPR	15.493293	8	23.753575	13
scran	15.493293	8	23.753575	13
TSS	15.493293	8	23.753575	13
CSS_DrImpute	25.896496	15	23.332294	7
edgeR_DrImpute	26.833863	16	21.105533	1
DESeq2_DrImpute	26.922529	17	26.652015	31
GMPR_DrImpute	27.666003	18	26.408340	30
scran_DrImpute	27.845489	19	25.762215	29
TSS_DrImpute	31.247418	20	21.650224	2
None_DrImpute	32.056846	21	25.115171	27
None_zCompositions_CZM	68.935412	22	31.893563	33
GMPR_zCompositions_CZM	69.064224	23	32.851038	34
DESeq2_zCompositions_CZM	69.091680	24	33.316633	36
scran_LowRank	69.472827	25	35.835972	38
scran_zCompositions_CZM	69.485258	26	31.238599	32
edgeR_LowRank	69.691890	27	56.932290	41
TSS_LowRank	69.946986	28	63.376871	42
CSS_LowRank	70.063393	29	48.875060	40
CSS_zCompositions_CZM	70.130414	30	25.202004	28
None_LowRank	70.252287	31	38.623301	39
GMPR_LowRank	70.610428	32	32.860882	35
DESeq2_LowRank	70.644355	33	33.326584	37
None_zCompositions_SQ	71.111817	34	23.583036	12
DESeq2_zCompositions_SQ	71.172882	35	23.362139	8
GMPR_zCompositions_SQ	71.179194	36	23.388271	9
CSS_zCompositions_SQ	71.686095	37	23.508952	11
scran_zCompositions_SQ	71.890836	38	23.413237	10
TSS_zCompositions_CZM	72.091026	39	23.102794	4
edgeR_zCompositions_CZM	72.331882	40	23.132833	5
TSS_zCompositions_SQ	72.340282	41	23.231930	6
edgeR_zCompositions_SQ	72.391466	42	23.051019	3
edgeR_LLSimpute	77.374670	43	81.044738	43
DESeq2_LLSimpute	78.496042	44	82.723031	48
scran_LLSimpute	79.551451	45	81.342883	44
CSS_LLSimpute	80.079156	46	82.840802	49
None_LLSimpute	80.079156	46	82.002482	47
TSS_LLSimpute	80.870712	48	81.793442	46
GMPR_LLSimpute	83.179420	49	81.469118	45

Table B.6 Simulated Dataset 3 - Pipelines median SMAPE and Aitchison's distance to the ground truth and their ranking.

Pipeline	SMAPE	Rank (SMAPE)	Aitchison dist	Rank (Aitchison dist)
Raw	2.717173	1	4.972023	1
CSS	2.717173	1	4.972023	1
DESeq2	2.717173	1	4.972023	1
edgeR	2.717173	1	4.972023	1
GMPR	2.717173	1	4.972023	1
scran	2.717173	1	4.972023	1
TSS	2.717173	1	4.972023	1
GMPR_scImpute	6.155422	8	25.240852	35
None_scImpute	6.157322	9	25.238466	29
DESeq2_scImpute	6.157767	10	25.240268	33
edgeR_scImpute	6.215741	11	25.239855	30
TSS_scImpute	6.218190	12	25.239993	32
scran_scImpute	6.219756	13	25.239884	31
CSS_scImpute	6.221034	14	25.240414	34
TSS_DrImpute	13.164630	15	15.219066	22
edgeR_DrImpute	13.196997	16	15.274801	23
CSS_DrImpute	16.065117	17	19.695972	24
GMPR_DrImpute	16.197855	18	22.286694	28
None_DrImpute	16.215554	19	22.201202	27
DESeq2_DrImpute	16.279755	20	21.002240	26
scran_DrImpute	16.320865	21	19.953838	25
TSS_LLSimpute	77.105263	22	51.852426	36
DESeq2_LLSimpute	77.982249	23	55.867086	40
None_LLSimpute	78.508772	24	54.432876	39
GMPR_LLSimpute	78.665752	25	58.037479	41
CSS_LLSimpute	78.839477	26	62.896008	42
scran_LLSimpute	79.736842	27	53.877309	38
edgeR_LLSimpute	80.769573	28	53.554698	37
TSS_LowRank	88.916205	29	84.310294	48
edgeR_LowRank	92.258819	30	84.528488	49
GMPR_LowRank	94.046151	31	74.082139	47
CSS_LowRank	94.495924	32	72.803843	45
DESeq2_LowRank	94.833300	33	68.920629	44
scran_LowRank	95.038494	34	67.389097	43
TSS_zCompositions_CZM	95.467177	35	5.185448	8
edgeR_zCompositions_CZM	95.487342	36	5.474512	9
TSS_zCompositions_SQ	95.531891	37	10.120804	14
CSS_zCompositions_CZM	95.539286	38	7.328895	10
CSS_zCompositions_SQ	95.568275	39	10.635669	15
scran_zCompositions_SQ	95.573044	40	11.544971	17
edgeR_zCompositions_SQ	95.583867	41	12.489447	18
scran_zCompositions_CZM	95.619631	42	8.061627	11
None_zCompositions_CZM	95.620942	43	8.940612	12
None_zCompositions_SQ	95.649147	44	13.788496	20
DESeq2_zCompositions_SQ	95.674623	45	12.856930	19
GMPR_zCompositions_SQ	95.675906	46	15.097380	21
DESeq2_zCompositions_CZM	95.697123	47	9.219752	13
GMPR_zCompositions_CZM	95.816378	48	11.366359	16
None_LowRank	95.919080	49	73.498354	46

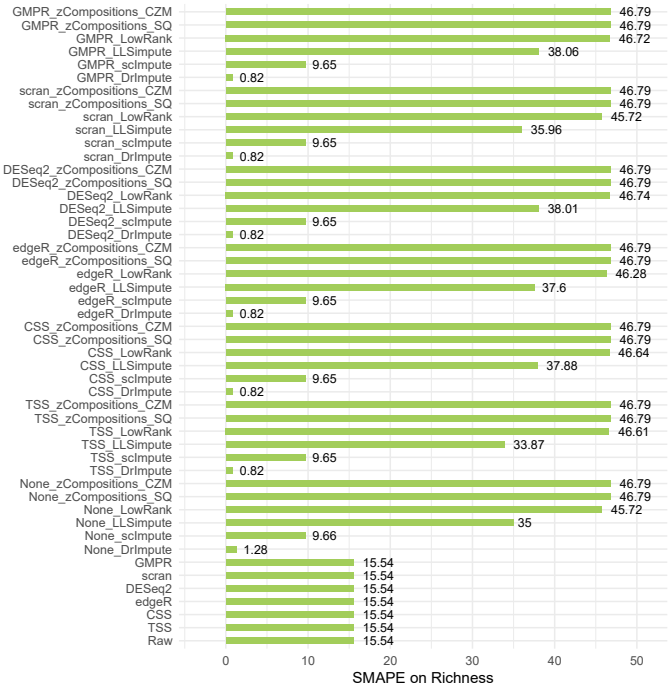


Fig. B.2 Simulated Dataset 1 - SMAPE between richness index calculated on real and on raw and pre-processed data.

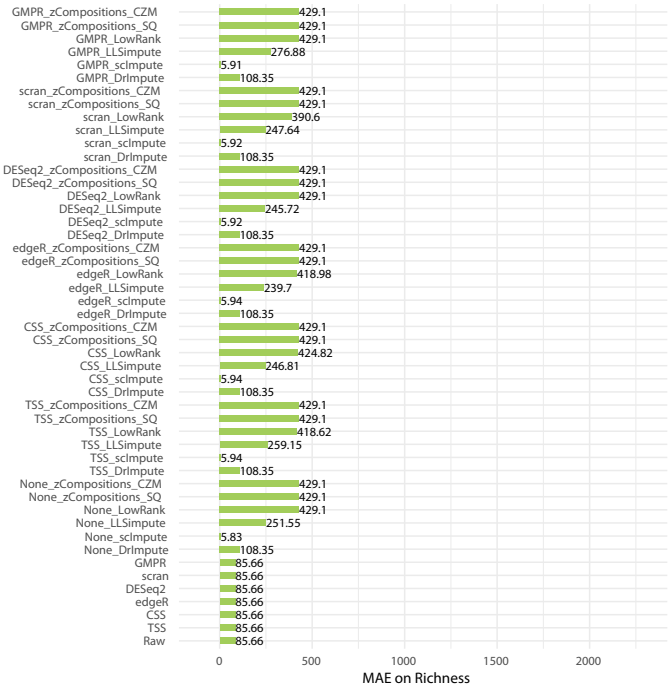


Fig. B.3 Simulated Dataset 2 - MAE between richness index calculated on real and on raw and pre-processed data.

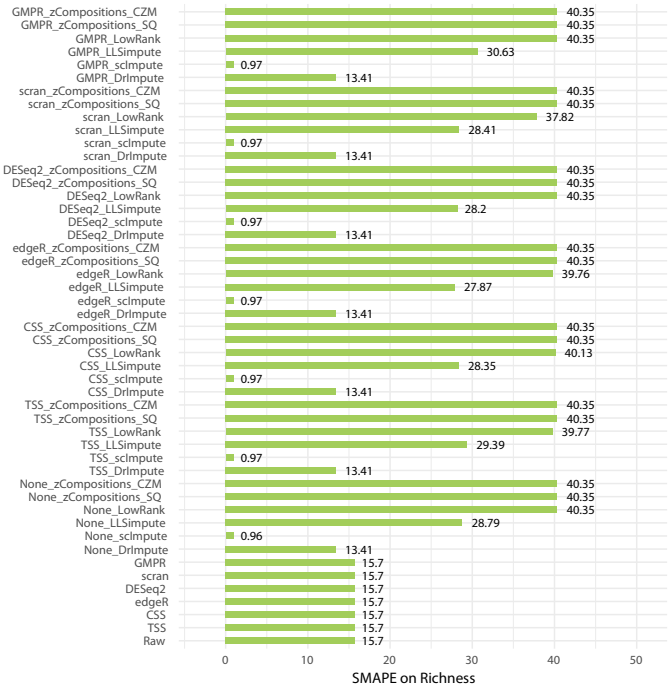


Fig. B.4 Simulated Dataset 2 - SMAPE between richness index calculated on real and on raw and pre-processed data.

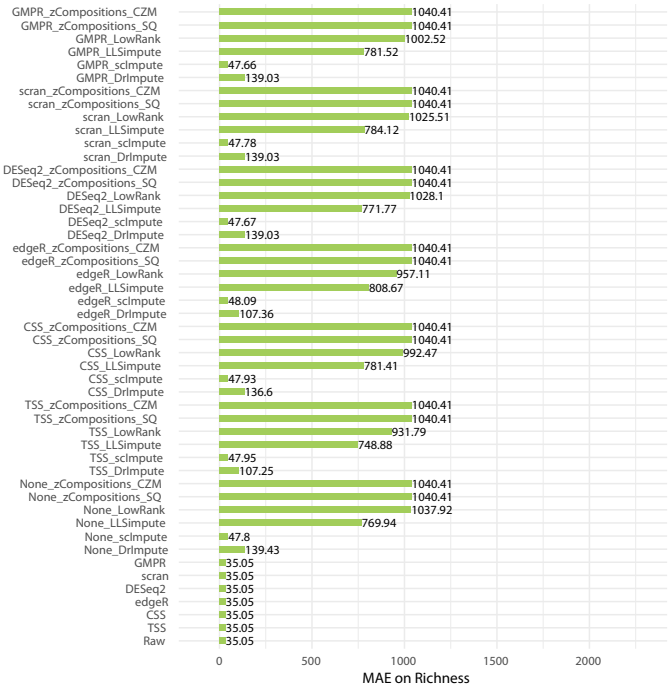


Fig. B.5 Simulated Dataset 3 - MAE between richness index calculated on real and on raw and pre-processed data.

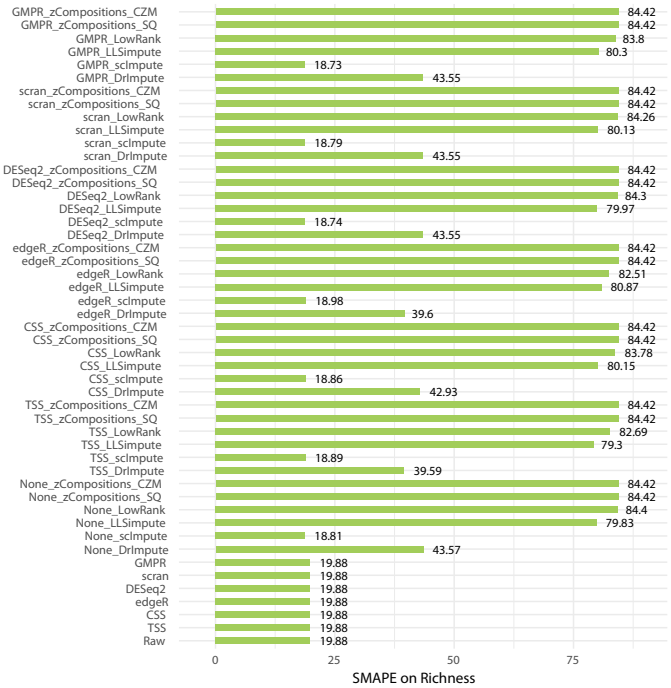


Fig. B.6 Simulated Dataset 3 - SMAPE between richness index calculated on real and on raw and pre-processed data.

Table B.7 Simulated Dataset 1 - Results of the pipelines on alpha indices of diversity. In the "Wrong comparisons" column is reported, in decreasing order for each index, the number of group-group comparisons not agreeing with the ground truth. The percentage over the total number of comparisons is also reported.

Tail			*Shannon*			*iSimpson*		
Pipeline	Wrong comparisons	%	Pipeline	Wrong comparisons	%	Pipeline	Wrong comparisons	%
TSS_DrImpute	2	2.20	Raw	0	0.00	Raw	0	0.00
scran_DrImpute	2	2.20	TSS	0	0.00	TSS	0	0.00
TSS_zCompositions_SQ	3	3.30	CSS	0	0.00	CSS	0	0.00
TSS_zCompositions_CZM	3	3.30	edgeR	0	0.00	edgeR	0	0.00
GMPR_DrImpute	3	3.30	DESeq2	0	0.00	DESeq2	0	0.00
Raw	4	4.40	scran	0	0.00	scran	0	0.00
TSS	4	4.40	GMPR	0	0.00	GMPR	0	0.00
CSS	4	4.40	None_zCompositions_SQ	0	0.00	None_DrImpute	0	0.00
edgeR	4	4.40	TSS_zCompositions_SQ	0	0.00	None_zCompositions_SQ	0	0.00
DESeq2	4	4.40	TSS_zCompositions_CZM	0	0.00	None_zCompositions_CZM	0	0.00
scran	4	4.40	CSS_zCompositions_SQ	0	0.00	TSS_DrImpute	0	0.00
GMPR	4	4.40	CSS_zCompositions_CZM	0	0.00	TSS_zCompositions_SQ	0	0.00
None_DrImpute	4	4.40	edgeR_zCompositions_SQ	0	0.00	TSS_zCompositions_CZM	0	0.00
None_scImpute	4	4.40	DESeq2_zCompositions_SQ	0	0.00	CSS_DrImpute	0	0.00
None_zCompositions_SQ	4	4.40	DESeq2_zCompositions_CZM	0	0.00	CSS_zCompositions_SQ	0	0.00
TSS_scImpute	4	4.40	scran_zCompositions_SQ	0	0.00	CSS_zCompositions_CZM	0	0.00
CSS_scImpute	4	4.40	GMPR_zCompositions_SQ	0	0.00	edgeR_DrImpute	0	0.00
CSS_zCompositions_SQ	4	4.40	GMPR_zCompositions_CZM	0	0.00	edgeR_zCompositions_SQ	0	0.00
edgeR_scImpute	4	4.40	None_scImpute	1	1.10	edgeR_zCompositions_CZM	0	0.00
edgeR_zCompositions_SQ	4	4.40	None_zCompositions_CZM	1	1.10	DESeq2_DrImpute	0	0.00
DESeq2_DrImpute	4	4.40	TSS_scImpute	1	1.10	DESeq2_zCompositions_SQ	0	0.00
DESeq2_scImpute	4	4.40	CSS_scImpute	1	1.10	DESeq2_zCompositions_CZM	0	0.00
DESeq2_zCompositions_SQ	4	4.40	edgeR_scImpute	1	1.10	scran_DrImpute	0	0.00
scran_scImpute	4	4.40	edgeR_zCompositions_CZM	1	1.10	scran_zCompositions_SQ	0	0.00
scran_zCompositions_SQ	4	4.40	DESeq2_scImpute	1	1.10	scran_zCompositions_CZM	0	0.00
GMPR_scImpute	4	4.40	scran_scImpute	1	1.10	GMPR_DrImpute	0	0.00
GMPR_zCompositions_SQ	4	4.40	scran_zCompositions_CZM	1	1.10	GMPR_zCompositions_SQ	0	0.00
CSS_DrImpute	5	5.49	GMPR_scImpute	1	1.10	GMPR_zCompositions_CZM	0	0.00
edgeR_DrImpute	5	5.49	TSS_DrImpute	2	2.20	None_scImpute	1	1.10
scran_LowRank	9	9.89	CSS_DrImpute	2	2.20	TSS_scImpute	1	1.10
scran_zCompositions_CZM	11	12.09	edgeR_DrImpute	2	2.20	CSS_scImpute	1	1.10
GMPR_zCompositions_CZM	11	12.09	scran_LowRank	2	2.20	edgeR_scImpute	1	1.10
DESeq2_zCompositions_CZM	13	14.29	DESeq2_DrImpute	3	3.30	DESeq2_scImpute	1	1.10
CSS_zCompositions_CZM	14	15.38	scran_DrImpute	3	3.30	scran_scImpute	1	1.10
edgeR_zCompositions_CZM	20	21.98	GMPR_DrImpute	3	3.30	GMPR_scImpute	1	1.10
None_zCompositions_CZM	24	26.37	None_DrImpute	5	5.49	scran_LowRank	2	2.20
None_LLSimpute	30	32.97	GMPR_LowRank	5	5.49	GMPR_LowRank	3	3.30
None_LowRank	30	32.97	CSS_LowRank	6	6.59	CSS_LowRank	4	4.40
GMPR_LowRank	32	35.16	DESeq2_LowRank	6	6.59	DESeq2_LowRank	7	7.69
DESeq2_LowRank	33	36.26	edgeR_LowRank	9	9.89	edgeR_LowRank	8	8.79
CSS_LowRank	38	41.76	None_LowRank	10	10.99	None_LowRank	11	12.09
scran_LLSimpute	42	46.15	TSS_LowRank	10	10.99	TSS_LowRank	13	14.29
TSS_LowRank	48	52.75	None_LLSimpute	43	47.25	None_LLSimpute	46	50.55
edgeR_LLSimpute	48	52.75	edgeR_LLSimpute	52	57.14	TSS_LLSimpute	46	50.55
edgeR_LowRank	48	52.75	scran_LLSimpute	52	57.14	edgeR_LLSimpute	49	53.85
CSS_LLSimpute	52	57.14	GMPR_LLSimpute	52	57.14	DESeq2_LLSimpute	54	59.34
TSS_LLSimpute	56	61.54	DESeq2_LLSimpute	53	58.24	CSS_LLSimpute	55	60.44
DESeq2_LLSimpute	61	67.03	CSS_LLSimpute	55	60.44	GMPR_LLSimpute	55	60.44
GMPR_LLSimpute	61	67.03	TSS_LLSimpute	65	71.43	scran_LLSimpute	58	63.74

Table B.8 Simulated Dataset 2 - Results of the pipelines on alpha indices of diversity. In the "Wrong comparisons" column is reported, in decreasing order for each index, the number of group-group comparisons not agreeing with the ground truth. The percentage over the total number of comparisons is also reported.

Tail			*Shannon*			*iSimpson*		
Pipeline	**Wrong comparisons**	**%**	**Pipeline**	**Wrong comparisons**	**%**	**Pipeline**	**Wrong comparisons**	**%**
TSS_DrImpute	1	3,57	*Raw*	0	0,00	*Raw*	0	0,00
edgeR_DrImpute	1	3,57	*TSS*	0	0,00	*TSS*	0	0,00
None_scImpute	2	7,14	*CSS*	0	0,00	*CSS*	0	0,00
TSS_scImpute	2	7,14	*edgeR*	0	0,00	*edgeR*	0	0,00
CSS_scImpute	2	7,14	*DESeq2*	0	0,00	*DESeq2*	0	0,00
edgeR_scImpute	2	7,14	*scran*	0	0,00	*scran*	0	0,00
DESeq2_scImpute	2	7,14	*GMPR*	0	0,00	*GMPR*	0	0,00
scran_scImpute	2	7,14	*None_LowRank*	0	0,00	*None_DrImpute*	0	0,00
scran_LowRank	2	7,14	*None_zCompositions_SQ*	0	0,00	*None_scImpute*	0	0,00
GMPR_scImpute	2	7,14	*None_zCompositions_CZM*	0	0,00	*None_LowRank*	0	0,00
Raw	3	10,71	*TSS_DrImpute*	0	0,00	*None_zCompositions_SQ*	0	0,00
TSS	3	10,71	*TSS_zCompositions_SQ*	0	0,00	*None_zCompositions_CZM*	0	0,00
CSS	3	10,71	*TSS_zCompositions_CZM*	0	0,00	*TSS_DrImpute*	0	0,00
edgeR	3	10,71	*CSS_DrImpute*	0	0,00	*TSS_scImpute*	0	0,00
DESeq2	3	10,71	*CSS_zCompositions_SQ*	0	0,00	*TSS_zCompositions_SQ*	0	0,00
scran	3	10,71	*CSS_zCompositions_CZM*	0	0,00	*TSS_zCompositions_CZM*	0	0,00
GMPR	3	10,71	*edgeR_DrImpute*	0	0,00	*CSS_DrImpute*	0	0,00
TSS_zCompositions_CZM	3	10,71	*edgeR_zCompositions_SQ*	0	0,00	*CSS_scImpute*	0	0,00
CSS_DrImpute	3	10,71	*edgeR_zCompositions_CZM*	0	0,00	*CSS_LowRank*	0	0,00
edgeR_zCompositions_CZM	3	10,71	*DESeq2_LowRank*	0	0,00	*CSS_zCompositions_SQ*	0	0,00
DESeq2_LowRank	3	10,71	*DESeq2_zCompositions_SQ*	0	0,00	*CSS_zCompositions_CZM*	0	0,00
None_zCompositions_SQ	4	14,29	*DESeq2_zCompositions_CZM*	0	0,00	*edgeR_DrImpute*	0	0,00
TSS_zCompositions_SQ	4	14,29	*scran_LowRank*	0	0,00	*edgeR_scImpute*	0	0,00
CSS_zCompositions_SQ	4	14,29	*scran_zCompositions_SQ*	0	0,00	*edgeR_zCompositions_SQ*	0	0,00
edgeR_zCompositions_SQ	4	14,29	*scran_zCompositions_CZM*	0	0,00	*edgeR_zCompositions_CZM*	0	0,00
DESeq2_zCompositions_SQ	4	14,29	*GMPR_DrImpute*	0	0,00	*DESeq2_DrImpute*	0	0,00
scran_DrImpute	4	14,29	*GMPR_LowRank*	0	0,00	*DESeq2_scImpute*	0	0,00
scran_zCompositions_SQ	4	14,29	*GMPR_zCompositions_SQ*	0	0,00	*DESeq2_LowRank*	0	0,00
GMPR_zCompositions_SQ	4	14,29	*GMPR_zCompositions_CZM*	0	0,00	*DESeq2_zCompositions_SQ*	0	0,00
GMPR_LowRank	5	17,86	*DESeq2_DrImpute*	1	3,57	*DESeq2_zCompositions_CZM*	0	0,00
DESeq2_DrImpute	6	21,43	*scran_DrImpute*	1	3,57	*scran_DrImpute*	0	0,00
GMPR_zCompositions_CZM	6	21,43	*None_DrImpute*	2	7,14	*scran_scImpute*	0	0,00
None_DrImpute	7	25,00	*None_scImpute*	2	7,14	*scran_LowRank*	0	0,00
None_LowRank	7	25,00	*TSS_scImpute*	2	7,14	*scran_zCompositions_SQ*	0	0,00
None_zCompositions_CZM	7	25,00	*TSS_LowRank*	2	7,14	*scran_zCompositions_CZM*	0	0,00
CSS_zCompositions_CZM	7	25,00	*CSS_scImpute*	2	7,14	*GMPR_DrImpute*	0	0,00
scran_zCompositions_CZM	7	25,00	*CSS_LowRank*	2	7,14	*GMPR_scImpute*	0	0,00
GMPR_DrImpute	7	25,00	*edgeR_scImpute*	2	7,14	*GMPR_LowRank*	0	0,00
DESeq2_zCompositions_CZM	10	35,71	*DESeq2_scImpute*	2	7,14	*GMPR_zCompositions_SQ*	0	0,00
edgeR_LLSimpute	14	50,00	*scran_scImpute*	2	7,14	*GMPR_zCompositions_CZM*	0	0,00
CSS_LowRank	16	57,14	*GMPR_scImpute*	2	7,14	*TSS_LowRank*	1	3,57
None_LLSimpute	17	60,71	*edgeR_LowRank*	4	14,29	*edgeR_LowRank*	3	10,71
CSS_LLSimpute	18	64,29	*None_LLSimpute*	12	42,86	*None_LLSimpute*	12	42,86
GMPR_LLSimpute	18	64,29	*CSS_LLSimpute*	14	50,00	*scran_LLSimpute*	12	42,86
TSS_LowRank	19	67,86	*edgeR_LLSimpute*	17	60,71	*CSS_LLSimpute*	14	50,00
scran_LLSimpute	22	78,57	*scran_LLSimpute*	18	64,29	*GMPR_LLSimpute*	17	60,71
edgeR_LowRank	23	82,14	*GMPR_LLSimpute*	18	64,29	*TSS_LLSimpute*	19	67,86
DESeq2_LLSimpute	24	85,71	*TSS_LLSimpute*	20	71,43	*edgeR_LLSimpute*	19	67,86
TSS_LLSimpute	25	89,29	*DESeq2_LLSimpute*	20	71,43	*DESeq2_LLSimpute*	21	75,00

Table B.9 Simulated Dataset 3 - Results of the pipelines on alpha indices of diversity. In the "Wrong comparisons" column is reported, in decreasing order for each index, the number of group-group comparisons not agreeing with the ground truth. The percentage over the total number of comparisons is also reported.

Tail			***Shannon***			***iSimpson***		
Pipeline	**Wrong comparisons**	**%**	**Pipeline**	**Wrong comparisons**	**%**	**Pipeline**	**Wrong comparisons**	**%**
Raw	0	0.00	*Raw*	0	0.00	*Raw*	0	0.00
TSS	0	0.00	*TSS*	0	0.00	*TSS*	0	0.00
CSS	0	0.00	*CSS*	0	0.00	*CSS*	0	0.00
edgeR	0	0.00	*edgeR*	0	0.00	*edgeR*	0	0.00
DESeq2	0	0.00	*DESeq2*	0	0.00	*DESeq2*	0	0.00
scran	0	0.00	*scran*	0	0.00	*scran*	0	0.00
GMPR	0	0.00	*GMPR*	0	0.00	*GMPR*	0	0.00
TSS_zCompositions_SQ	0	0.00	*None_zCompositions_SQ*	0	0.00	*None_zCompositions_SQ*	0	0.00
TSS_zCompositions_CZM	0	0.00	*None_zCompositions_CZM*	0	0.00	*None_zCompositions_CZM*	0	0.00
edgeR_zCompositions_SQ	0	0.00	*TSS_zCompositions_SQ*	0	0.00	*TSS_DrImpute*	0	0.00
scran_zCompositions_SQ	1	1.52	*TSS_zCompositions_CZM*	0	0.00	*TSS_zCompositions_SQ*	0	0.00
TSS_DrImpute	2	3.03	*CSS_zCompositions_SQ*	0	0.00	*TSS_zCompositions_CZM*	0	0.00
edgeR_DrImpute	2	3.03	*edgeR_DrImpute*	0	0.00	*CSS_DrImpute*	0	0.00
DESeq2_zCompositions_SQ	2	3.03	*edgeR_zCompositions_SQ*	0	0.00	*CSS_zCompositions_SQ*	0	0.00
scran_DrImpute	3	4.55	*edgeR_zCompositions_CZM*	0	0.00	*CSS_zCompositions_CZM*	0	0.00
CSS_zCompositions_SQ	4	6.06	*DESeq2_zCompositions_SQ*	0	0.00	*edgeR_zCompositions_SQ*	0	0.00
None_DrImpute	5	7.58	*DESeq2_zCompositions_CZM*	0	0.00	*edgeR_zCompositions_CZM*	0	0.00
None_zCompositions_SQ	5	7.58	*scran_zCompositions_SQ*	0	0.00	*DESeq2_zCompositions_SQ*	0	0.00
DESeq2_DrImpute	5	7.58	*GMPR_zCompositions_SQ*	0	0.00	*DESeq2_zCompositions_CZM*	0	0.00
GMPR_DrImpute	5	7.58	*None_DrImpute*	1	1.52	*scran_zCompositions_SQ*	0	0.00
None_scImpute	6	9.09	*TSS_DrImpute*	1	1.52	*scran_zCompositions_CZM*	0	0.00
TSS_scImpute	6	9.09	*CSS_zCompositions_CZM*	1	1.52	*GMPR_DrImpute*	0	0.00
CSS_scImpute	6	9.09	*scran_zCompositions_CZM*	1	1.52	*GMPR_zCompositions_SQ*	0	0.00
edgeR_scImpute	6	9.09	*GMPR_DrImpute*	1	1.52	*GMPR_zCompositions_CZM*	0	0.00
DESeq2_scImpute	6	9.09	*GMPR_zCompositions_CZM*	1	1.52	*None_DrImpute*	1	1.52
scran_scImpute	6	9.09	*scran_DrImpute*	2	3.03	*edgeR_DrImpute*	1	1.52
GMPR_scImpute	6	9.09	*CSS_DrImpute*	3	4.55	*DESeq2_DrImpute*	1	1.52
CSS_DrImpute	7	10.61	*DESeq2_DrImpute*	3	4.55	*scran_DrImpute*	1	1.52
GMPR_zCompositions_SQ	7	10.61	*None_scImpute*	12	18.18	*DESeq2_LowRank*	6	9.09
scran_zCompositions_CZM	19	28.79	*TSS_scImpute*	12	18.18	*GMPR_LowRank*	6	9.09
CSS_zCompositions_CZM	22	33.33	*CSS_scImpute*	12	18.18	*None_LowRank*	7	10.61
DESeq2_zCompositions_CZM	25	37.88	*edgeR_scImpute*	12	18.18	*None_scImpute*	10	15.15
scran_LLSimpute	29	43.94	*DESeq2_scImpute*	12	18.18	*TSS_scImpute*	10	15.15
CSS_LLSimpute	32	48.48	*scran_scImpute*	12	18.18	*CSS_scImpute*	10	15.15
edgeR_zCompositions_CZM	34	51.52	*GMPR_scImpute*	12	18.18	*CSS_LowRank*	10	15.15
GMPR_zCompositions_CZM	41	62.12	*DESeq2_LowRank*	17	25.76	*edgeR_scImpute*	10	15.15
edgeR_LowRank	46	69.70	*scran_LowRank*	19	28.79	*DESeq2_scImpute*	10	15.15
TSS_LowRank	48	72.73	*scran_LLSimpute*	22	33.33	*scran_scImpute*	10	15.15
DESeq2_LLSimpute	48	72.73	*GMPR_LowRank*	22	33.33	*scran_LowRank*	10	15.15
CSS_LowRank	49	74.24	*None_LowRank*	24	36.36	*GMPR_scImpute*	10	15.15
edgeR_LLSimpute	49	74.24	*CSS_LowRank*	24	36.36	*TSS_LowRank*	27	40.91
None_zCompositions_CZM	50	75.76	*TSS_LowRank*	39	59.09	*edgeR_LowRank*	32	48.48
GMPR_LLSimpute	50	75.76	*edgeR_LLSimpute*	39	59.09	*scran_LLSimpute*	32	48.48
GMPR_LowRank	53	80.30	*DESeq2_LLSimpute*	39	59.09	*None_LLSimpute*	38	57.58
None_LLSimpute	54	81.82	*CSS_LLSimpute*	41	62.12	*DESeq2_LLSimpute*	38	57.58
DESeq2_LowRank	54	81.82	*edgeR_LowRank*	45	68.18	*CSS_LLSimpute*	39	59.09
scran_LowRank	57	86.36	*GMPR_LLSimpute*	51	77.27	*TSS_LLSimpute*	44	66.67
None_LowRank	58	87.88	*None_LLSimpute*	56	84.85	*edgeR_LLSimpute*	45	68.18
TSS_LLSimpute	62	93.94	*TSS_LLSimpute*	56	84.85	*GMPR_LLSimpute*	49	74.24

References

[1] Antoni van Leeuwenhoek. *The select works of anthony van leeuwenhoek: containing his microscopical discoveries in many of the works of nature*, volume 1. translator, 1800.

[2] Carl R Woese and George E Fox. Phylogenetic structure of the prokaryotic domain: the primary kingdoms. *Proceedings of the National Academy of Sciences*, 74(11):5088–5090, 1977.

[3] Christoph A Thaiss, Maayan Levy, Tal Korem, Lenka Dohnalová, Hagit Shapiro, Diego A Jaitin, Eyal David, Deborah R Winter, Meital Gury-BenAri, Evgeny Tatirovsky, and others. Microbiota diurnal rhythmicity programs host transcriptome oscillations. *Cell*, 167(6):1495–1510, 2016.

[4] Brandilyn A Peters, Jean A Shapiro, Timothy R Church, George Miller, Chau Trinh-Shevrin, Elizabeth Yuen, Charles Friedlander, Richard B Hayes, and Jiyoung Ahn. A taxonomic signature of obesity in a large study of American adults. *Scientific reports*, 8(1):9749, 2018.

[5] Merly Escalona, Sara Rocha, and David Posada. A comparison of tools for the simulation of genomic next-generation sequencing data. *Nature reviews. Genetics*, 17(8):459–469, 8 2016.

[6] Florent E. Angly, Dana Willner, Forest Rohwer, Philip Hugenholtz, and Gene W. Tyson. Grinder: A versatile amplicon and shotgun sequence simulator. *Nucleic Acids Research*, 40(12), 2012.

[7] Kerensa E. McElroy, Fabio Luciani, and Torsten Thomas. GemSIM: General, error-model based simulator of next-generation sequencing data. *BMC Genomics*, 13(1), 2012.

[8] Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, Richard Durbin, and 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, 8 2009.

[9] Daniel C Richter, Felix Ott, Alexander F Auch, Ramona Schmid, and Daniel H Huson. MetaSim—A Sequencing Simulator for Genomics and Metagenomics. *PLoS ONE*, 3(10):e3373, 10 2008.

[10] Hao Wu, Chi Wang, and Zhijin Wu. PROPER: Comprehensive power evaluation for differential expression using RNA-seq. *Bioinformatics*, 31(2):233–241, 2015.

[11] Ran Bi and Peng Liu. ssizeRNA: Sample Size Calculation for RNA-Seq Experimental Design, 2017.

[12] Juhee Lee and Marilou Sison-mangus. A Bayesian Semiparametric Regression Model for Joint Analysis of Microbiome Data. 9(March):1–14, 2018.

[13] Mark D Robinson and Gordon K Smyth. Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, 23(21):2881–2887, 11 2007.

[14] Francis H C Crick. On protein synthesis. In *Symp Soc Exp Biol*, volume 12, page 8, 1958.

[15] J D Watson and F H C Crick. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, 171:737, 4 1953.

[16] Fred Sanger and Alan R Coulson. A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of molecular biology*, 94(3):441–448, 1975.

[17] Frederick Sanger, Gilian M Air, Bart G Barrell, Nigel L Brown, Alan R Coulson, John C Fiddes, Clyde A Hutchison III, Patrick M Slocombe, and Mo Smith. Nucleotide sequence of bacteriophage φX174 DNA. *nature*, 265(5596):687, 1977.

[18] Frederick Sanger, Steven Nicklen, and Alan R Coulson. DNA sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences*, 74(12):5463–5467, 1977.

[19] Howard M Temin, S Mizutami, and others. RNA-dependent DNA polymerase in virions of Rous sarcoma virus. *Nature*, 226:1211–1213, 1970.

[20] David Baltimore. Viral RNA-dependent DNA polymerase: RNA-dependent DNA polymerase in virions of RNA tumour viruses. *Nature*, 226(5252):1209, 1970.

[21] Jerzy K Kulski ED1 Jerzy K Kulski. Next-Generation Sequencing — An Overview of the History, Tools, and "Omic" Applications. page Ch. 1. IntechOpen, Rijeka, 2016.

[22] J Craig Venter, Mark D Adams, Eugene W Myers, Peter W Li, Richard J Mural, Granger G Sutton, Hamilton O Smith, Mark Yandell, Cheryl A Evans, Robert A Holt, and others. The sequence of the human genome. *science*, 291(5507):1304–1351, 2001.

[23] International Human Genome Sequencing Consortium and others. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860, 2001.

[24] Tong Zhu. Global analysis of gene expression using GeneChip microarrays. *Current opinion in plant biology*, 6(5):418–425, 2003.

[25] Tim Lenoir and Eric Giannella. The emergence and diffusion of DNA microarray technology. *Journal of biomedical discovery and collaboration*, 1(1):11, 2006.

[26] Lilian T C França, Emanuel Carrilho, and Tarso B L Kist. A review of DNA sequencing techniques. *Quarterly reviews of biophysics*, 35(2):169–200, 2002.

[27] K Wetterstrand. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP).

[28] Sara Goodwin, John D McPherson, and W Richard McCombie. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics*, 17(6):333, 2016.

[29] Elaine R Mardis. A decade's perspective on DNA sequencing technology. *Nature*, 470(7333):198, 2011.

[30] Elaine R Mardis. Next-generation DNA sequencing methods. *Annu. Rev. Genomics Hum. Genet.*, 9:387–402, 2008.

[31] Michael L Metzker. Sequencing technologies—the next generation. *Nature reviews genetics*, 11(1):31, 2010.

[32] Marcel Margulies, Michael Egholm, William E Altman, Said Attiya, Joel S Bader, Lisa A Bemben, Jan Berka, Michael S Braverman, Yi-Ju Chen, Zhoutao Chen, and others. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376, 2005.

[33] Shankar Balasubramanian. Solexa sequencing: decoding genomes on a population scale. *Clinical chemistry*, 61(1):21–24, 2015.

[34] Rajiv C McCoy, Ryan W Taylor, Timothy A Blauwkamp, Joanna L Kelley, Michael Kertesz, Dmitry Pushkarev, Dmitri A Petrov, and Anna-Sophie Fiston-Lavier. Illumina TruSeq synthetic long-reads empower de novo assembly and resolve complex, highly-repetitive transposable elements. *PloS one*, 9(9):e106689, 2014.

[35] Jonathan M Rothberg, Wolfgang Hinz, Todd M Rearick, Jonathan Schultz, William Mileski, Mel Davey, John H Leamon, Kim Johnson, Mark J Milgrew, Matthew Edwards, and others. An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, 475(7356):348, 2011.

[36] Kevin J Travers, Chen-Shan Chin, David R Rank, John S Eid, and Stephen W Turner. A flexible and efficient template format for circular consensus sequencing and SNP detection. *Nucleic acids research*, 38(15):e159–e159, 2010.

[37] Hagan Bayley. Nanopore sequencing: from imagination to reality. *Clinical chemistry*, 61(1):25–31, 2015.

[38] David Stoddart, Andrew J Heron, Ellina Mikhailova, Giovanni Maglia, and Hagan Bayley. Single-nucleotide discrimination in immobilized DNA oligonucleotides with a biological nanopore. *Proceedings of the National Academy of Sciences*, 106(19):7702–7707, 2009.

[39] Michael Begon, John L Harper, Colin R Townsend, and others. *Ecology. Individuals, populations and communities.* Blackwell scientific publications, 1986.

[40] Alejandra Escobar-Zepeda, Arturo de León, and Alejandro Sanchez-Flores. The road to metagenomics: from microbiology to DNA sequencing technologies and bioinformatics. *Frontiers in genetics*, 6:348, 2015.

[41] Julian R Marchesi and Jacques Ravel. The vocabulary of microbiome research: a proposal, 2015.

[42] Ilaria Laudadio, Valerio Fulci, Francesca Palone, Laura Stronati, Salvatore Cucchiara, and Claudia Carissimi. Quantitative Assessment of Shotgun Metagenomics and 16S rDNA Amplicon Sequencing in the Study of Human Gut Microbiome. *Omics: a journal of integrative biology*, 22(4):248–254, 2018.

[43] Anniina Rintala, Sami Pietilä, Eveliina Munukka, Erkki Eerola, Juha-Pekka Pursiheimo, Asta Laiho, Satu Pekkala, and Pentti Huovinen. Gut microbiota analysis results are highly dependent on the 16S rRNA gene target region, whereas the impact of DNA extraction is minor. *Journal of biomolecular techniques: JBT*, 28(1):19, 2017.

[44] Adam G Clooney, Fiona Fouhy, Roy D Sleator, Aisling O'Driscoll, Catherine Stanton, Paul D Cotter, and Marcus J Claesson. Comparing apples and oranges?: next generation sequencing and its impact on microbiome analysis. *PLoS One*, 11(2):e0148028, 2016.

[45] Masayasu Nomura, Richard Gourse, and Gail Baughman. Regulation of the synthesis of ribosomes and ribosomal components. *Annual review of biochemistry*, 53(1):75–117, 1984.

[46] Juliane C Dohm, Claudio Lottaz, Tatiana Borodina, and Heinz Himmelbauer. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic acids research*, 36(16):e105, 2008.

[47] R P Smyth, T E Schlub, A Grimm, V Venturi, A Chopra, S Mallal, M P Davenport, and J Mak. Reducing chimera formation during PCR amplification to ensure accurate genotyping. *Gene*, 469(1):45–51, 2010.

[48] Zongzhi Liu, Todd Z DeSantis, Gary L Andersen, and Rob Knight. Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers. *Nucleic acids research*, 36(18):e120–e120, 2008.

[49] Yijun Sun, Yunpeng Cai, Li Liu, Fahong Yu, Michael L Farrell, William McKendree, and William Farmerie. ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences. *Nucleic acids research*, 37(10):e76–e76, 2009.

[50] Patrick D Schloss and Sarah L Westcott. Assessing and improving methods used in OTU-based approaches for 16S rRNA gene sequence analysis. *Applied and environmental microbiology*, 2011.

[51] J Gregory Caporaso, Justin Kuczynski, Jesse Stombaugh, Kyle Bittinger, Frederic D Bushman, Elizabeth K Costello, Noah Fierer, Antonio Gonzalez Pena, Julia K Goodrich, Jeffrey I Gordon, and others. QIIME allows analysis of high-throughput community sequencing data. *Nature methods*, 7(5):335, 2010.

[52] Wei Chen, Clarence K Zhang, Yongmei Cheng, Shaowu Zhang, and Hongyu Zhao. A comparison of methods for clustering 16S rRNA sequences into OTUs. *PloS one*, 8(8):e70837, 2013.

[53] Xiaolin Hao, Rui Jiang, and Ting Chen. Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering. *Bioinformatics*, 27(5):611–618, 2011.

[54] Benjamin J Callahan, Paul J McMurdie, and Susan P Holmes. Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *The ISME journal*, 11(12):2639, 2017.

[55] Benjamin J Callahan, Paul J McMurdie, Michael J Rosen, Andrew W Han, Amy Jo A Johnson, and Susan P Holmes. DADA2: high-resolution sample inference from Illumina amplicon data. *Nature methods*, 13(7):581, 2016.

[56] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied and Environmental Microbiology*, 72(7):5069–5072, 2006.

[57] Bonnie L Maidak, Gary J Olsen, Niels Larsen, Ross Overbeek, Michael J McCaughey, and Carl R Woese. The ribosomal database project (RDP). *Nucleic acids research*, 24(1):82–85, 1996.

[58] Christian Quast, Elmar Pruesse, Pelin Yilmaz, Jan Gerken, Timmy Schweer, Pablo Yarza, Jörg Peplies, and Frank Oliver Glöckner. The SILVA ribosomal RNA gene database project: Improved data processing and web-based tools. *Nucleic Acids Research*, 41(D1):590–596, 2013.

[59] Sophie Weiss, Zhenjiang Zech Xu, Shyamal Peddada, Amnon Amir, Kyle Bittinger, Antonio Gonzalez, Catherine Lozupone, Jesse R. Zaneveld, Yoshiki Vázquez-Baeza, Amanda Birmingham, Embriette R. Hyde, and Rob Knight. Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome*, 5(1):27, 2017.

[60] Lizhen Xu, Andrew D Paterson, Williams Turpin, and Wei Xu. Assessment and Selection of Competing Models for Zero-Inflated Microbiome Data. *PLOS ONE*, 10(7):e0129606, 7 2015.

[61] Gregory B Gloor, Jean M Macklaim, Vera Pawlowsky-Glahn, and Juan J Egozcue. Microbiome Datasets Are Compositional: And This Is Not Optional. *Frontiers in Microbiology*, 8:2224, 11 2017.

[62] J.Aitchison. The Statistical Analysis of Compositional data. 44(2):139–177, 1986.

[63] Vera Pawlowsky-Glahn and Antonella Buccianti. *Compositional Data Analysis: Theory and Applications*. John Wiley & Sons, 2011.

[64] Vera Pawlowsky-Glahn and Antonella Buccianti. *Compositional data analysis: Theory and applications*. John Wiley & Sons, 2011.

[65] Juan José Egozcue, Vera Pawlowsky-Glahn, Glòria Mateu-Figueras, and Carles Barcelo-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

[66] John Aitchison. The statistical analysis of compositional data. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 139–177, 1982.

[67] Siddhartha Mandal, Will Van Treuren, Richard A. White, Merete Eggesbø, Rob Knight, and Shyamal D. Peddada. Analysis of composition of microbiomes: a novel method for studying microbial composition. *Microbial Ecology in Health & Disease*, 26(0):1–7, 2015.

[68] Ionas Erb, Thomas Quinn, David Lovell, and Cedric Notredame. Differential proportionality – a normalization-free approach to differential gene expression. *Proceedings of CoDaWork 2017, the 7th Compositional Data Analysis Workshop, Abbadia San Salvatore, Italy.*, pages 1–14, 2017.

[69] John C Marioni, Christopher E Mason, Shrikant M Mane, Matthew Stephens, and Yoav Gilad. RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18(9):1509–1517, 9 2008.

[70] Joseph N. Paulson, O. Colin Stine, Héctor Corrada Bravo, and Mihai Pop. Differential abundance analysis for microbial marker-gene surveys. *Nature Methods*, 10(12):1200–1202, 2013.

[71] Joseph Nathaniel Paulson, M Pop, and H C Bravo. metagenomeSeq: Statistical analysis for sparse high-throughput sequencing. *Bioconductor package*, 1(0), 2013.

[72] Mark D Robinson, Davis J McCarthy, and Gordon K Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 1 2010.

[73] Michael I. Love, Wolfgang Huber, and Simon Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, 15(12):1–21, 2014.

[74] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106, 2010.

[75] Aaron T.L. Lun, Karsten Bach, and John C. Marioni. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome Biology*, 17(1):1–14, 2016.

[76] Li Chen, James Reeve, Lujun Zhang, Shengbing Huang, Xuefeng Wang, and Jun Chen. GMPR: A robust normalization method for zero-inflated count data with application to microbiome sequencing data. *PeerJ*, 6:e4600, 2018.

[77] Wuming Gong, Il-Youp Kwak, Pruthvi Pota, Naoko Koyano-Nakagawa, and Daniel J Garry. DrImpute: imputing dropout events in single cell RNA sequencing data. *BMC Bioinformatics*, 19(1):220, 2018.

[78] Wei Vivian Li and Jingyi Jessica Li. An accurate and robust imputation method scImpute for single-cell RNA-seq data. *Nature communications*, 9(1):997, 2018.

[79] Hyunsoo Kim, Gene H. Golub, and Haesun Park. Missing value estimation for DNA microarray gene expression data: Local least squares imputation. *Bioinformatics*, 21(2):187–198, 2005.

[80] C. Chen, B. He, and X. Yuan. Matrix completion via an alternating direction method. *IMA Journal of Numerical Analysis*, 32(1):227–245, 2012.

[81] Lihua Zhang and Shihua Zhang. Comparison of computational methods for imputing single-cell RNA-sequencing data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2018.

[82] Javier Palarea-Albaladejo and Josep Antoni Martín-Fernández. ZCompositions - R package for multivariate imputation of left-censored data under a compositional approach. *Chemometrics and Intelligent Laboratory Systems*, 143:85–96, 2015.

[83] Josep-Antoni Martín-Fernández, Karel Hron, Matthias Templ, Peter Filzmoser, and Javier Palarea-Albaladejo. Bayesian-multiplicative treatment of count zeros in compositional data sets. *Statistical Modelling: An International Journal*, 15(2):134–158, 2015.

[84] Josep Daunis-i estadella, Josep Antoni Martín-Fernández, and Javier Palarea-Albaladejo. Bayesian tools for count zeros in compositional data sets. *Analysis*.

[85] Josep A Martín-Fernández, Carles Barceló-Vidal, and Vera Pawlowsky-Glahn. Dealing with zeros and missing values in compositional data sets using nonparametric imputation. *Mathematical Geology*, 35(3):253–278, 2003.

[86] Matthias Templ, Karel Hron, and Peter Filzmoser. robCompositions: An R-package for Robust Statistical Analysis of Compositional Data. page 341–355, 2011.

[87] K. Hron, M. Templ, and P. Filzmoser. Imputation of missing values for compositional data using classical and robust methods. *Computational Statistics and Data Analysis*, 54(12):3095–3107, 2010.

[88] J Aitchison, C Barceló-Vidal, J A Mart\'\in-Fernández, and V Pawlowsky-Glahn. Logratio analysis and compositional distance. *Mathematical Geology*, 32(3):271–275, 2000.

[89] The Human Microbiome Project Consortium. A framework for human microbiome research. *Nature*, 486(7402):215–221, 2012.

[90] The Human Microbiome Project Consortium. Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–214, 2012.

[91] Anna Klindworth, Elmar Pruesse, Timmy Schweer, Jörg Peplies, Christian Quast, Matthias Horn, and Frank Oliver Glöckner. Evaluation of general 16S ribosomal RNA gene PCR primers for classical and next-generation sequencing-based diversity studies. *Nucleic Acids Research*, 41(1):1–11, 2013.

[92] Simon Andrews. FastQC: a quality control tool for high throughput sequence data, 2010.

[93] Guido van Rossum. Python Reference Manual. *CWI Report CS-R9525*, 1995.

[94] J. G. Caporaso, C. L. Lauber, W. A. Walters, D. Berg-Lyons, C. A. Lozupone, P. J. Turnbaugh, N. Fierer, and R. Knight. Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. *Proceedings of the National Academy of Sciences*, 108(Supplement_1):4516–4522, 2011.

[95] Robert C. Edgar. UPARSE: Highly accurate OTU sequences from microbial amplicon reads. *Nature Methods*, 10(10):996–998, 2013.

[96] Leho Tedersoo, Tom W. May, and Matthew E. Smith. Ectomycorrhizal lifestyle in fungi: Global diversity, distribution, and evolution of phylogenetic lineages. *Mycorrhiza*, 20(4):217–263, 2010.

[97] Gary Xie, Chien-Chi Lo, Matthew Scholz, and Patrick S G Chain. Recruiting human microbiome shotgun data to site-specific reference genomes. *PloS one*, 9(1):e84963, 2014.

[98] Diane Lambert. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, 34(1):1–14, 1992.

[99] John Mullahy. Specification and testing of some modified count data models. *Journal of econometrics*, 33(3):341–365, 1986.

[100] Julia W. Neilson, Katy Califf, Cesar Cardona, Audrey Copeland, Will van Treuren, Karen L. Josephson, Rob Knight, Jack A. Gilbert, Jay Quade, J. Gregory Caporaso, and Raina M. Maier. Significant Impacts of Increasing Aridity on the Arid Soil Microbiome. *mSystems*, 2(3):00195–16, 2017.

[101] Author M B Wilk and R Gnanadesikan. Biometrika Trust Probability Plotting Methods for the Analysis of Data Published by : Oxford University Press on behalf of Biometrika Trust Stable URL : http://www.jstor.org/stable/2334448. 55(1):1–17, 2017.

[102] John W Tukey. *Exploratory data analysis*, volume 2. Reading, Mass., 1977.

[103] Nathaniel Phillips. yarrr: A Companion to the e-Book "YaRrr!: The Pirate's Guide to R", 2017.

[104] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[105] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 1977.

[106] Gail M. Sullivan and Richard Feinn. Using Effect Size - or Why the P Value Is Not Enough. *Journal of Graduate Medical Education*, 4(3):279–282, 2012.

[107] Ruth Cano-Corres, Javier Sánchez-Álvarez, and Xavier Fuentes-Arderiu. The Effect Size: Beyond Statistical Significance. *Ejifcc*, 23(1):19–23, 2012.

[108] Robert Coe. It's the effect size, stupid. What effect size is and why it is important. *British Educational Research Association annual conference*, pages 1–18, 2002.

[109] Shlomo S. Sawilowsky. New Effect Size Rules of Thumb. *Journal of Modern Applied Statistical Methods*, 8(2):597–599, 2009.

[110] Francesca Finotello, Eleonora Mastrorilli, and Barbara Di Camillo. Measuring the diversity of the human microbiota with targeted next-generation sequencing. *Briefings in Bioinformatics*, (September 2016):bbw119, 2016.

[111] R H Whittaker. Vegetation of the Siskiyou Mountains, Oregon and California. *Ecological Monographs*, 30(3):279–338, 1960.

[112] Robert H Whittaker. Evolution and measurement of species diversity. *Taxon*, pages 213–251, 1972.

[113] Hanna Tuomisto. A diversity of beta diversities: straightening up a concept gone awry. Part 1. Defining beta diversity as a function of alpha and gamma diversity. *Ecography*, 33(1):2–22, 2010.

[114] Lou Jost. Partitioning diversity into independent alpha and beta components. *Ecology*, 88(10):2427–2439, 2007.

[115] Anne Chao. Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, pages 783–791, 1987.

[116] Eric P Smith and Gerald van Belle. Nonparametric estimation of species richness. *Biometrics*, pages 119–129, 1984.

[117] Mark O Hill. Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2):427–432, 1973.

[118] E Kathryn Morris, Tancredi Caruso, François Buscot, Markus Fischer, Christine Hancock, Tanja S Maier, Torsten Meiners, Caroline Müller, Elisabeth Obermaier, Daniel Prati, and others. Choosing and using diversity indices: insights for ecological applications from the German Biodiversity Exploratories. *Ecology and evolution*, 4(18):3514–3524, 2014.

[119] Kelvin Li, Monika Bihan, Shibu Yooseph, and Barbara A Methe. Analyses of the microbial diversity across the human microbiome. *PloS one*, 7(6):e32118, 2012.

[120] Jari Oksanen, F Guillaume Blanchet, Michael Friendly, Roeland Kindt, Pierre Legendre, Dan McGlinn, Peter R Minchin, R B O'Hara, Gavin L Simpson, Peter Solymos, M Henry H Stevens, Eduard Szoecs, and Helene Wagner. vegan: Community Ecology Package, 2018.

[121] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)*, pages 289–300, 1995.

[122] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin del la Société Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[123] Gail M Sullivan and Richard Feinn. Using effect size—or why the P value is not enough. *Journal of graduate medical education*, 4(3):279–282, 2012.