

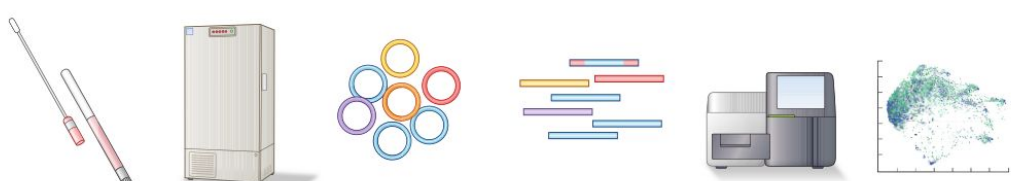
Frequently asked questions on microbial diversity analysis **(based on 16S rRNA marker gene sequencing)**

Belén Carbonetto, Bioinformatics Unit, Instituto Gulbenkian de Ciência,
Fundação Calouste Gulbenkian, 2780-156 Oeiras, Portugal

The study of microbial communities based on the analysis of the 16S rRNA marker gene usually includes the following steps [1]:

- 1) Sample collection
- 2) Sample transport and storage
- 3) DNA extraction
- 4) Library preparation (16S rRNA gene amplification)
- 5) Sequencing
- 6) Computational analysis

Every one of these steps may include bias and errors [2]. Figure 1 illustrates the main sources.



Sample processing step	Sample collection	Sample storage	DNA extraction	Sequencing library preparation	DNA sequencing	Computational analysis
Technical sources of error and bias	<ul style="list-style-type: none"> Inadequate sampling Incomplete sample stabilization Sampling kit contamination Mislabeled samples 	<ul style="list-style-type: none"> Change in community structure due to differential growth Degradation of DNA during freeze-thaw cycles 	<ul style="list-style-type: none"> Differential recovery of DNA from different strains Extraction kit contamination Sample swaps during transfer Sample cross-contamination 	<ul style="list-style-type: none"> Quantitative amplification bias (PCR efficiency) Qualitative amplification bias (primer mismatches) Amplification errors (PCR chimeras, substitution errors) Reagent contamination Sample cross-contamination 	<ul style="list-style-type: none"> Sequencing errors Run-to-run carryover Barcode swapping Demultiplexing errors 	<ul style="list-style-type: none"> Suboptimal quality control or filtering Alignment errors Database errors Database bias Batch effects Failure to flag contaminants

Figure 1: Sources of error and bias during each step of microbial diversity analysis based on 16S rRNA marker gene. Adapted from Gohl 2017 [3]

It is important to take all these sources of variation into account when defining the experimental design in order to minimize biases and technical errors.

As regards the computational and numerical analysis the following steps are needed:

6.1) Pre-process sequences: remove primers and barcodes, demultiplex samples and perform quality filtering.

- 6.2) Pick features and representative sequences
- 6.3) Assign taxonomy to reads or features
- 6.4) Align sequences and infer phylogeny
- 6.5) Calculate alpha and beta diversity
- 6.6) Test for differential abundances between groups of samples
- 6.7) Visualization

As shown in Figure 1 biases can be introduced during sequence processing and data analysis mainly related to the references and databases used. Moreover, several tools and algorithms are available for each step and different choices may lead to diverging final results and conclusions. It can be a daunting task to decide the optimal approach for analysis of a specific dataset. The choice of the best option will mainly depend on the characteristics of the data and several problems are related to the resolution of each step leading to the following list of “frequently asked questions” that will help decide which is the best option for each case.

Moreover taking this into account, we have decided in the UBI at IGC and for the ONEIDA users, the following pipeline for sequence processing and analysis:

[QIIME2](#) tools are used for sequence processing and numerical analysis:

- 0.1. **DADA2** is used for denoising, quality filtering, merging forward and reverse reads and feature (ASV) picking. Feature tables are filtered to remove singletons and doubletons.
- 0.2. **MAFFT** is used to align the representative sequences and later infer phylogenetic distances between features.
- 0.3. **Diversity analysis:** feature tables and phylogenetic information are then used to calculate alpha and beta diversity:

Alpha diversity metrics:

- Shannon’s diversity index (a quantitative measure of community richness)
- Observed OTUs (community richness)
- Faith’s Phylogenetic Diversity (a qualitative measure, i.e. presence/absence, of community richness that incorporates phylogenetic relationships between the features)
- Evenness (or Pielou’s Evenness; a measure of community evenness)

Beta diversity metrics:

- Jaccard distance (a qualitative measure of community dissimilarity)
- Bray-Curtis distance (a quantitative measure of community dissimilarity)
- Unweighted UniFrac distance (a qualitative measure of community dissimilarity that incorporates phylogenetic relationships between the features)
- Weighted UniFrac distance (a quantitative measure of community dissimilarity that incorporates phylogenetic relationships between the features)

Rarefaction is used to even sample sizes to the size of the smallest sample. Alpha diversity comparisons are done using Kruskal-Wallis test. Beta diversity comparisons: are done using Permanova, ANOSIM and Kruskal-Wallis test on distances.

2.4 Taxonomy classification is done using a bayesian classifier (similar to RDP classifier) trained with the Silva v.132 -99% classifier for 515F/806R region.

2.5 Differential abundance analysis on features or taxa is performed using ANCOM.

A guide through this steps can be found [here](#). This is the tutorial we give on the training sessions by the UBI (only the merging step is missing in the tutorial).

3. Numerical analysis and visualization using R scripts: Principal coordinate analysis is used to visualize beta diversity. Permdisp test is calculated to compare beta diversity dispersion. In order to analyse differential abundances of taxa between groups of samples we also use Mann Whitney and Kruskal Wallis tests on relative abundances of reads with FDR p-value adjustment. Moreover, pairwise comparisons are done using Conover test with FDR Benjamini-Hochberg correction when factors with more than 2 levels are evaluated. Boxplots and barplots are used for differential abundance visualization. Other custom visualizations can be done on request, too.

Frequently asked questions:

Question #1: Should controls and mock communities be added during sequencing?

The problem:

Even if this question is related with a decision that needs to be done before sequencing and it is not related with the choice of analysis methods we still decided to include this topic here and to discuss the advantages of using controls and mock communities.

The introduction of controls during the sequencing step will allow to identify the presence of deviations from the normal error rate and will lead to better decisions regarding data analysis. As mentioned, many sources of bias have been identified in 16S rRNA based studies: DNA extraction protocol, sequencing artifacts, DNA copy number, sampling depth, and primer design among the most important [4][5]. Even if it is not possible to control for every source of error and bias, bech recommendations to reduce them include:

- to perform multiple DNA extractions for each biological sample,
- to use pooled DNA from multiple extractions as PCR template for each sample,
- to use pooled PCR products from multiple amplification reactions for each library &
- to reduce the number of PCR cycles to avoid chimera formation

Quality filtering and trimming will allow to remove part of the errors, mainly systematic sequencing errors and chimeras (see next question for more details). However, random unknown bias due to alterations in an independent single run, or contaminations will not be detected by any of these methods.

Answer:

The use of mock samples as positive controls can help to identify problems between different runs. The introduction of mock communities as standards is the only way to verify “normal behaviour” of each run [6]. Ignoring that possibility may lead to serious errors that may obscure real patterns, or create false ones and lead to erroneous conclusions.

By using positive controls or mock communities you can account for the variation in the error distribution. For example, in the [mothur](#) platform [7] the *seq.error* command is included to calculate sequencing error rates based on mock community information by comparing mock reads to a reference fasta file. This, of course, requires one or more mock community samples of known composition to be sequenced and an accurate reference. Error rate is defined as: *Sum of mismatches to reference / Sum of bases in query*. Using the same mock community (i.e. same composition, same DNA extraction protocol, same PCR conditions) in every run, a distribution of errors can be calculated and “non- normal” deviations can be detected. Even if not calculating the error rate, it is good practice to add mock communities as standards and compare the expected number of species vs the observed number of OTUs (features) and to analyze the variation between runs.

Schirmer et.al. (2015) have analyzed error patterns on 16S rRNA amplicon data sequenced using MiSeq [8]. They showed that the library preparation method and the choice of primers are the most significant sources of bias. They showed that systematic errors can

be identified by the inference of individual error profiles for different sequencers, library preparation methods and sequencing types. If systematic errors can be detected, then adding mock communities in every run will allow the detection of non-systematic errors and eventually enable the correction of biases.

An alternative to the use of mock communities are Spike-in standards, which are relatively well established in the field of RNA-seq [9]. Tourlousse et. al. (2017) proposed the use of synthetic standards that represent artificial 16S rRNA genes with *in silico* designed variable regions with no identity to nucleotide sequences in public databases. Spike-in sequences can serve as ground truth to measure the accuracy and reproducibility of sequencing. These standards are intended to be employed as mixtures of multiple plasmid DNAs at a range of concentrations. They are then used as templates for amplification and library preparation in the same way as samples. As mock communities do, these standards can serve as references for estimating error rates.

The recommendation is to include mock communities in the sequencing run when possible, especially for projects that will include several sequencing runs. The sequencing facilities should include standard mock communities for a follow up of bias between runs.

The addition of negative controls is also recommended. “Blank controls” for DNA extraction and PCR reactions should be included. Even if there is not yet a state of the art method to deal with contaminating reads, the R package *decontam* is nowadays the most promising tool [10]. It is based on common signatures of contamination: i) sequences from contaminating taxa are likely to have frequencies that inversely correlate with sample DNA concentration and ii) sequences from contaminating taxa are likely to have higher prevalence in control samples than in true samples. However, the removal of contaminants should be evaluated for each case. How to handle potential contamination in downstream analyses will depend on the source of contamination, the extent of contamination, and whether the contaminating taxa likely overlap with the taxa expected to see in the samples. Blindly removing taxa from a taxon table will absolutely cause issues and artifacts downstream. For example, reagent contamination where few specific taxa are abundant in negative controls can be relatively easy to handle by identifying and removing those contaminant reads. However, if the contamination is a product of sample cross-over, simply removing all of the taxa that are abundant in the negative controls could lead to the removal of abundant taxa found in the actual samples. Likewise, the threshold for determining if contamination in the negative controls is problematic will depend on the read coverage across the sample set (e.g. if the negative controls have only ~100 reads, and your samples are >100,000 reads – it is unlikely to have serious contamination problems). Approaching contamination during data analysis cannot and should not be done automatically. Dealing with contamination must be done with the same level of care and thoughtfulness that would go into analyzing the actual biological data. Relying on bioinformatic automatization or random cutoffs to salvage contaminated datasets is sketch. It is a better choice to reduce the contamination in the first place. We recommend to read [this](#) blog post for further discussion.

Question #2: Should I filter data to reduce errors? How?

The problem:

Sources of mistake, bias and error during microbial diversity analysis are present in almost every step of processing and analysis as described above. (For a discussion on contamination handling please take a look at question #1.)

It is well known that Illumina sequencing approaches introduce errors [8]. Increased error rates are observed towards the end of the reads when using MiSeq technology. This is assumed to be due to accumulation of phasing and pre-phasing events throughout the sequencing process. Every time a molecule fails to elongate properly or advances too fast, the overall signal for the cluster suffers from interference. So as the read length increases, the cluster signal can get weaker due to an accumulation of these events resulting in higher error rates towards the end of the read [11]. Illumina systems provide Phred quality scores for every nucleotide, which represent the probability that a given base call is erroneous.

Filtering of data can also be performed at the “table level” after OTU/feature calling. Some methods are known to overestimate the number of OTUs present in a sample and filtering of the OTU table can be a good way to correct part of this problem (read question #4 for further details).

Answer:

The most widely used strategy for quality filtering works on a per-nucleotide basis, truncating reads at the position where their quality begins to drop [12]. Usually a minimum Phred quality score of 20 is chosen for read truncation at the 3' end of the reads. In other words, truncation at a read length when the Phred quality scores drops below Q20 (at the 25th percentile) is recommended. This will lead to reads with a probability of incorrect base calling of 1 in 100 (i.e. 99% accuracy). This quality truncation step is performed after primer and barcoding removal but before OTU or ASV picking.

Filtering strategies can be applied not only on sequence data but OTU tables can be filtered prior to diversity analysis. This is good practice when using methods that overestimate OTU number. Bokulich et al. (2013) [12] have suggested to remove OTUs that represent less than 0.005% of the total read abundance. Another common practice is to remove singletons and doubletons by sample or by table (i.e. OTUs with a count of one or two reads in the entire dataset) [13][14]. If a mock community was sequenced, it can be used as reference to define a threshold for removal of low count or low proportion OTUs [15].

The recommendation is: always filter/truncate reads according to a >Q20 threshold and eliminate low abundance OTUs from tables when using a method with known overestimation issues. Bokulich suggestion should be strict enough but if the use of this filter is not chosen, at least singletons and doubletons should be removed.

Question #3: Should I merge forward and reverse reads? Which method should I use?

The problem:

The MiSeq Illumina platform provides paired-end sequencing, in which a DNA sequence is read from both ends up to a specified read length. Currently the maximum read length provided by MiSeq is up to 300 bases. A target segment can result in an overlap between forward and reverse reads, which is usually the case of 16S rRNA gene amplicons, if primers were chosen correctly. Moreover, since quality tends to degrade towards the ends of the reads, reliable merging of overlapping paired-end reads can result in a combined DNA sequence that might permit bioinformatics correction of these 3'-end sequencing errors and yield higher quality sequence output. So even if only forward reads could be used for 16S rRNA gene based microbial diversity analysis, the addition of a merging step can yield to higher quality data, and sometimes even to longer reads. The proper combination of sequence quality filtering and read truncation will also define the quality of the final merging product.

Answer:

To find an answer to the question of which merger to use to join 16S rRNA gene Illumina forward and reverse reads, Samuel Dias Rosa Viana, a former master student in the IGC-UBI benchmarked several mergers [16]. He used two mock communities : the dataset known as "mock_1" in Kozich et. al. (2013) [17] with 21 bacteria species in even concentrations, already used in several benchmark studies [18][19], and an IGC custom made mock community with 8 bacterial species at different concentrations. He tested the following mergers: BBMerge, Fastq-Join, Flash, the *make.contigs* command as implemented in mothur, PandaSeq, Pear and USEARCH. He evaluated precision and recall based on ground truth defined by the number of mismatches of the alignment of the forward reads using Blast against the reference genome. If the number of mismatches after Blast alignment of the merged reads was equal or lower than for the forward read it was assigned as a true positive. If the number of mismatches increased, it was assigned as false positive. True negatives were defined by reads rejected for merging and false negatives by reads that were not merged but should have been merged. Results showed that besides BBMerge, who performed the worst, the rest of the mergers had comparable performance. Fastq-Join and FLASH showed the highest precision, along with PEAR and Usearch. He also observed that trimming for quality prior to merging improved recall but lowered precision. Other authors have also compared the performance of mergers obtaining similar results [20]. They have shown that fastq-join was capable of operating at high level of specificity while still maintaining a low false negative rate.

The tested merging methods are (most of) the possible choices before using a OTU picking method based on sequence similarity. However, if not defining OTUs but ASV (please see question #4) you may not need to choose a merger since the method provides a merging step (e.g DADA2 which does not support merged reads as input [18]).

Our recommendation in this case is linked with the answer to question #4, we choose to use the merging method included in DADA2 which merge forward and reverse reads after denoising. This is because DADA2 denoising algorithm uses the

empirical relationship between the quality score and the error rates. When reads are merged, this relationship will differ between the forward-only, overlapping, and reverse-only portions of the merged read. That variation interferes with the denoising algorithm, and therefore greater accuracy can be achieved by denoising before merging.

Question #4: Which is the best OTU picking method?

The problem:

The analysis of diversity based on 16S rRNA gene sequences begins with the construction of operational taxonomic units (OTUs): clusters of reads that differ by less than an arbitrary fixed sequence dissimilarity threshold, most commonly 3%. The clusterization of reads into OTUs is supposed to serve two purposes. First to translate taxonomic concepts developed in other systems into the context of marker-gene sequencing of microbial communities [21]. Second to reduce the impact of amplicon sequencing error on measures of diversity and community composition by grouping errors together with “error-free” sequences.

Approaches for OTU clustering can be mainly divided in two groups: **de novo** and **reference based** (a.k.a. **closed-reference**).

De novo OTUs are constructed by clustering reads that are sufficiently similar to one another. *De novo* OTUs are emergent features of a data set, with boundaries and membership that depend on the data set in which they are defined. As a consequence, the delineation of *de novo* OTUs depends on the relative abundances of the sampled community even in the limit of infinite sequencing depth and zero errors. *De novo* methods are usually computationally demanding with long runtimes. Within *de novo* methods we can find hierarchical algorithms such as nearest, furthest, and average neighbor [22] and algorithms that employ heuristic as implemented in USEARCH [22,23], Sumacust, OTUCLUST [24] and Swarm [25].

On the other side, closed-reference OTUs (also called phylotypes [26]) are properties of a reference database; each reference sequence in the database defines and labels an associated closed-reference OTU. Sequencing reads are assigned to a closed-reference OTUs if they are sufficiently similar to the associated reference sequence (usually 97% similarity). If the same reference database is used, closed-reference OTU assignments from independently processed data sets can be validly compared. However, biological variation that is not represented in the reference database is lost during assignment to closed-reference OTUs.

In sum the main problems of OTU picking strategies based on sequence similarity are:

- De novo* methods are supposed to be more informative but the runtime is high and usually cannot then be scaled to modern-sized data sets.
- Closed reference methods are easily parallelizable and runtime is acceptable, but the major drawback is that they cannot identify novel diversity: if a sequence has no match in the reference database, it cannot be included in the analysis, restricting analyses to already-known taxa .

In order to overcome these problems a dual method called **open-reference** OTU picking, has been proposed. This combines *de novo* and closed reference protocols. First, input sequences are clustered against a reference database in a closed-reference OTU picking process. However, rather than discarding sequences that fail to match the reference, these “failures” are clustered *de novo*. This method offers benefits over both the *de novo* and closed-reference protocols because it includes the parallel closed-reference step that it will typically run faster than *de novo* OTU picking and, since it includes *de novo* OTU picking of the sequences that fail to hit the reference database, all sequences are clustered, so analyses are not restricted to already-known OTUs. Mainly because of this reason, open-reference OTU picking has become the most used method for OTU calling mainly due to popularization by QIIME authors [27].

However, it is already known that OTU picking methods often output numbers of OTUs even over one order of magnitude higher than the number of species present in samples. This overestimation was observed when analysing mock communities and it is known to be related with sequencing errors, which cannot be accounted during OTU picking procedures. OTU overestimation is a known issue even when performing appropriate previous quality control and filtering. To overcome this, sub-OTU approaches have been proposed. The most popular **sub-OTUs** methods are Deblur [28], DADA2 [18] and UNOISE2 [29]. They attempt to obtain single-nucleotide resolution from Illumina data with statistical methods that infer the putative true “mother” sequences within a sample. This “mother” sequences gave rise to the distribution of observed error-prone sequences. This true sequences are called “**amplicon sequence variants**” (ASV). In other words, ASVs are inferred by a *de novo* process in which biological (true) sequences are discriminated from errors on the basis of, in part, the expectation that biological sequences are more likely to be repeatedly observed than are error-containing sequence.

In sum, there is a variety of methods for OTU or sub-OTU picking, each of them with their own drawbacks and advantages. Which one should we choose?

Answer:

Some benchmark studies have been done in order to address this question. For example, Kopylova et al. [22] performed a comprehensive benchmark in which they analyzed precision and recall of a list of OTU picking methods. They tested *de novo*: Swarm, Sumacust, UPARSE, UCLUST, OTUCLUST and *mothur's* average nearest, furthest, and average neighbor algorithms; closed- reference: USEARCH, SortMeRNA and UCLUST; and open-reference algorithms: UCLUST, SortMeRNA, Sumacust and USEARCH.

They tested the algorithms on several data sets: simulated, mock communities and environmental samples. Here we are interested in the tests performed on simulated and mock datasets because precision and recall can be measured in comparison to a ground truth table, which is not the case for environmental communities. They generated two 16S rRNA gene simulated datasets as fastq files. The first one called “sim_even” represents an even distribution of 1,076 species (mother sequences), randomly subsampled from the Greengenes 97% database and computationally amplified at the same depth and length by extracting the V4 region. They used ART simulator [30] for amplification and sequencing simulation. The second data set called “sim_staggered” represents the same 1,076 species

as the “sim_even” data set but amplified at different (random) species abundance level. As mock communities they have chosen the Bokulich_2, Bokulich_3, and Bokulich_6) from Bokulich et al. (2013) [12].

Precision and recall were calculated based on a ground truth of expected taxonomic composition (i.e. reads’ original taxonomic string stored in the simulated FASTA for simulated reads and the classification of the reference sequences using RDP classifier method, and the GreenGenes 13.8 database clustered at 97%). In order to simplify the visualization of results they used the F score to compare methods defined by:

$$F=2 \times \text{precision} \times \text{recall}/(\text{precision} + \text{recall})$$

Where F can take values from 0 to 1 and where 1 is the best possible score.

Results for simulated data are summarized in the following table:

Method	Software	sim_even		sim_staggered	
		OTUs	F-score	OTUs	F-score
<i>de novo</i>	swarm	1042	0.84	1035	0.83
	sumacust	1031	0.83	1022	0.83
	uparse_q3	1013	0.84	997	0.84
	uclust	1045	0.83	1035	0.83
	otuclust_q3	996	0.84	953	0.81
	mothur near neighbor	957	0.82	949	0.81
	mothur further neighbor	978	0.82	970	0.81
	mothur average neighbor	963	0.82	959	0.82
closed reference	uclust	1238	0.83	1225	0.84
	sortmerna	1072	0.82	1067	0.81
open reference	uclust	1262	0.83	1245	0.83
	sortmerna+ sumacust	1072	0.82	1085	0.81

Table 1: Results of OTU picking strategies on simulated datasets. “sim_even” dataset harbors of 1,076 species (i.e. distinct 16s rRNA gene V4 region sequences, randomly subsampled from the Greengenes 97% database and computationally amplified at the same depth and length). “sim_staggered” represents the same 1,076 species but amplified at different (random) species abundance level. Adapted from Kopylova et. al. (2016) [31].

The F value was almost the same for all methods. They found that Swarm, SUMACLUSt, UCLUSt, and UPARSE (with relaxed parameters) performed equally well on simulated datasets, with mothur_average and otuclust closely behind. Here we are more interested in results for the two most used methods: open-reference OTU with UCLUSt as implemented in QIIME and Average Linkage as implemented in *mothur*. Since the ground truth was 1,076 OTUs, we conclude that the average linkage method was underestimating the number of OTUs while UCLUSt was overestimating it.

For the mock communities, most tools were able to correctly detect the expected number and identity of genera, but only UPARSE reported significantly fewer false-positive taxa (followed by OTUCLUSt and USEARCH). However, all methods overestimated the number of OTUs over one order of magnitude. In terms of accurately predicted taxonomic

composition for de novo tools, Swarm performed well across all simulated and mock datasets, followed closely by SUMACLUSt and UCLUSt.

Recently a benchmark performed by Edgar [32] using a different approach showed similar results. In this case the test dataset was not simulated, he constructed a database of high-quality, full-length 16S rRNA sequences from known species. He used all prokaryotic genome assemblies in GenBank annotated as "Complete" in GenBank [33]. He selected one assembly at random for each species when more than one assembly is available, and trimmed for the V4 hypervariable region. By considering a completely different ground truth (i.e. a pair of sequences was correctly classified if they belong to the same species and were clustered in the same OTU), he measured the accuracy of OTU pickers as the correlation between correlation between OTUs and species. He evaluated the performance of *de novo* algorithms: nearest-neighbor average-neighbor, furthest-neighbor and OptiClust as implemented in *mothur*. He concluded that all algorithms achieved comparably high scores by a given quality and that no algorithm is intrinsically superior.

We are also interested in the comparison for sub-OTUS methods compared to OTU methods, so we ran our own comparisons in the UBI. We included the following OTUs or sub-OTUs pickers with default parameters:

- OPTICLUSt as implemented in *mothur*
- Average-neighbour method as implemented in *mothur*
- Deblur as implemented in QIIME2
- UCLUSt for open-reference OTU picking as implemented in QIIME

For UCLUSt as implemented in QIIME, sequences were previously quality filtered and denoised using commands *split_libraries_fastq.py* (with -q 19 and default parameters) and *identify_chimeric_seqs.py* with usearch method with default parameters. For Deblur, sequences were quality filtered using the QIIME2 command *quality-filter q-score* (with --p-quality-window 19 and default parameters). For average-neighbour method and OPTICLUSt sequences were quality filtered using the steps described [here](#).

The simulated datasets in Kopylova et. al. (2016), previously described above, were used. Taxonomic classification of representative sequences after OTU picking was done using RDP classifier with a cutoff of 80% using the QIIME command *assign_taxonomy.py* and the GreenGenes OTU database clustered at 97% similarity level. Ground truth taxonomy tables in Kopylova et. al. (2016) were used as expected taxa to calculate precision and recall. Moreover, the Jaccard index was used to quantify the level of similarity between the observed and expected values. This is possible because our observations are taxonomy strings which have a "semantic structure" (i.e. taxonomy is hierarchical and has a tree structure with branches and nodes, so it has a topology). It is then possible to define a measure of similarity between expected and observed taxonomy strings even if they were miss-classified. Expected and observed OTUs may not match the complete taxonomy string but may share ancestral nodes. We used Jaccard index which is the proportion of shared nodes between A and B relative to the total number of nodes connected to A or B. Results obtained are presented in Table 2.

Method	Precision	Recall	F-score	Jaccard Similarity	Observed OTUs
<i>mothur</i> - average neighbour	0.78	0.88	0.83	0.93	2,010
Deblur	0.83	0.88	0.85	0.94	1,057
Open reference UCLUST	0.82	0.88	0.85	0.94	1,253
<i>mothur</i> -OptiClust	0.79	0.84	0.82	0.91	1,294

Table 2: Results of OTU picking strategies on the “sim_even” dataset which harbors 1,076 species (i.e. distinct 16s rRNA gene V4 region sequences, randomly subsampled from the Greengenes 97% database and computationally amplified at the same depth and length).

Results were, as expected, similar to those presented by Kopylova et. al. As regards Deblur, not presented in the reviewed benchmark, we observed that it outperformed the other methods, it got the higher F-score and similarity value. Moreover, it was the method with closest number of observed OTUs compared to the expected (1,076).

Benjamin Callahan also performed this exact benchmark on DADA2. Results are not published but are available [here](#). His results showed that DADA2 was substantially more accurate than any of the methods tested in the Kopylova et al. paper, outperforming on both sensitivity and specificity. He also observed that the default QIIME/ucrust pipeline reported 20,084 OTUs in an 18-strain mock community, ~20,000 of which are spurious. In comparison, DADA2 reported just 39 sequence variants, which consist of the 18 expected strains, a handful of contaminants, and ~5 spurious sequences. This dramatic reduction in spurious output was achieved without imposing aggressive filters (i.e. OTU table filtering).

In sum, we recommend the use of ASV methods and in particular DADA2 to “cluster” sequences.

Question #5: Which is the best method for taxonomic classification? Which is the best reference taxonomy?

Taxonomic classification of 16S gene sequences typically requires comparing query sequences to annotated database sequences. There are two main approaches for this:

- 1) Pattern recognition algorithms such as **k-mer based methods**, e.g. the RDP Classifier [34] and SPINGO[35].

These methods compare the frequency of k-mer nucleotides between query and database sequences. The main advantage of k-mer based approaches is its fast

computational speed. However, k-mer based approaches rely on two key assumptions: i) the k-mer nucleotides in DNA sequences used as discriminating features among different taxa are independent, and ii) the actual nucleotide position of the k-mers in the DNA sequences is not important. In reality, nucleotides in different positions of a gene sequence can be correlated (e.g., to preserve the secondary or higher-dimensional structure of rRNA folding), and gene sequences with the same set of k-mer in different orders are clearly not the same sequences. Therefore, these two assumptions are the theoretical sources of taxonomic misclassification by k-mer based approaches. In summary, these approaches rely on a proxy measurement of the sequence similarity between the query and database sequences, which is inherently less accurate than the gold standard sequence-alignment-based method.

2) Methods based on **alignments** and reconstruction of **phylogenetic trees** (e.g. Blast based methods) [36].

The main problem with Blast is that it does not automatically provide a definitive taxonomic classification for the query sequence. Instead, Blast based methods produce an extensive list of hits and a robust algorithm is then required to analyze the report and extract the appropriate information that defines the taxonomy of the query at the deepest taxonomic rank possible. Criteria used in this final step will be critical for classification accuracy. The selection of the best hit is crucial. It is common to have several hits from different taxa that may have comparable sequence similarities to the query sequence. Therefore, it is not reliable to simply transfer the taxonomic annotation associated with the best database hit for the query sequence. It is required that one hit is significantly better than other in order to be selected. One of the most used methods to overcome this problem is The Lowest Common Ancestor (LCA) algorithm. For example, if a query sequence has two BLAST hits belonging to two different species, e.g., one from *Lactobacillus acidophilus* and the other one from *L. casei*, the LCA algorithm assigns the query sequence to the genus *Lactobacillus*, which is the lowest common taxonomic level of these two species.

Independently of the chosen approach, comparison of classifiers can be really challenging. The first main problem is that classifiers use different taxonomies which cannot be directly compared, e.g. NCBI, Silva [37] and Greengenes[38]. A second problem is that classifiers vary in the way they report confidence on the prediction.

Edgar benchmarked several methods including: RDP classifier with its own RDP training set, QIIME uclust-based classifier, Sortmerna and blast as implemented in QIIME against the last release of the GreenGenes reference database; and RDP as implemented in *mothur* against Silva database, among others [39]. Sensitivity was measured as the fraction of known queries that are correctly identified so that the highest achievable sensitivity by an ideal algorithm is 100%. If novel queries were also counted then sensitivity <100% would have reflected an opaque combination of low database coverage and failures to correctly predict known taxa. Two types of false positive were defined: i) misclassifications, where an incorrect name is predicted for a known taxa, and ii) over-classifications, where a name is predicted for unknown taxa. Taking this into account and for a given query set, a reference database and a taxonomic level he defined: N_{known} and N_{novel} as the number of queries with known and novel taxa respectively. Then true positives (TP) were defined as the number of

correct predictions and misclassification FP_{mis} as the number of misclassification errors and FP_{over} as the number of over-classification errors. The total number of queries was

$$N = N_{known} + N_{novel}$$

He then defined the following accuracy metrics:

$$\text{Sensitivity} = TP / N_{known}$$

$$\text{Misclassification rate} = MC = FP_{mis} / N_{known}$$

$$\text{Over-classification rate} = OC = FP_{over} / N_{novel}$$

$$\text{Errors per query} = EPQ = (FP_{mis} + FP_{over}) / N$$

The results of his benchmark revealed that the over-classification error was >10% for all evaluated combinations, being highest for the Blast based method (87.4%). Blast against GreenGenes database revealed the highest misclassification error rate (17.1%). RDP classifier using the RDP training set got the higher sensitivity score (94.0%), followed by Blast and RDP implemented in QIIME with 82.7% and 81.7% respectively. The default method for classification in *mothur* (i.e. RDP classifier with Silva database) performed better in terms of errors than the default method in QIIME (i.e. Uclust based classifier with GreenGenes database) with 30.8% and 48.4% OC rates and 2.4% and 9.7% MC rates, respectively. QIIME approach had higher sensitivity, though (77.3% vs 75.8%).

The most popular and available taxonomy databases (and used by Edgar in his benchmark) are:

- **RDP** [40], with 13,212 sequences belonging to 2,126 genera
- **SILVA** [40,41]; its version v123 has 1.8M small subunit ribosomal RNA sequences and v114 was estimated to contain ~94,000 genera [42].
- **Greengenes** [43], its version v13.5 has 1.8M 16S sequences.

Most taxonomy annotations in SILVA and Greengenes are predictions obtained by computational and manual analyses which are primarily based on trees predicted from multiple alignments; in the RDP training set most annotations were predicted using RDP classifier [38][44]. In further detail, the RDP database contains 16S rRNA sequences available from the International Nucleotide Sequence Database Collaboration (INSDC) databases [45]. Names of the organisms associated with the sequences are obtained as the most recently published synonym from [Bacterial Nomenclature Up-to-Date](#). Information on taxonomic classification for Bacteria and Archaea is based on the taxonomic roadmaps by [Bergey's Trust](#) and *List of Prokaryotic Names with Standing in Nomenclature* (LPSN) [46]. The GreenGenes classification is based on automatic *de novo* tree construction and rank mapping from other taxonomy sources (mainly NCBI). Phylogenetic tree is constructed from 16S rRNA sequences that have been obtained from public databases and passed a quality

filtering. Sequences are aligned by their characters and secondary structure and then subjected to tree construction with FastTree [47]. Inner nodes are automatically assigned taxonomic ranks from NCBI supplemented with previous version of Greengenes taxonomy and [CyanoDB](#). The SILVA database is based primarily on phylogenies for 16S rRNA and Taxonomic rank information for Archaea and Bacteria is obtained from *Bergey's Taxonomic Outlines* [48] and from the LPSN [46]. Taxonomic rank assignments in the SILVA database are manually curated.

A recently published benchmark extended Edgar's work to compare the state of the art classification methods [49]. Almeida et al. compared the default classifiers of MAPseq, mothur, QIIME, and QIIME 2 using synthetic simulated datasets comprised of some of the most abundant genera found in the human gut, ocean, and soil environments. They limited the analysis to classification at the lineage level instead of operational taxonomic units. They measured precision and recall using both GreenGenes v 13_8 and Silva v128 databases at the family and genus level.

They showed that all tools tested performed moderately well, with high precision and modest-to-high recall rates at the genus level. QIIME 2 presented significant improvements over the other tools, particularly over the preceding version of QIIME, in regard to detection sensitivity at both family and genus levels. Results support the use of QIIME 2 to obtain the largest proportion of classified sequences at the most accurate relative abundances. Nevertheless, the results also showed MAPseq to be a more conservative and precise approach, meaning that fewer genera were misassigned. In addition, this tool showed considerably better computational performance than QIIME 2, requiring approximately 30 times less memory and almost one-half the CPU time to process the same dataset.

They have also observed that the SILVA 128 database performed better than Greengenes 13_8 in terms of recall at both genus and family levels as well as in predicting the true taxa composition of the simulated communities. Nonetheless, there are additional advantages to the use of SILVA: it is more frequently updated (Greengenes was last updated in May 2013); it includes rRNA sequences of eukaryotic organisms in addition to archaea and bacterial species; and has been shown to be more easily comparable and mapped to other taxonomies such as the NCBI.

In sum, our recommendation is to use the RDP classifier approach in any of its implementations with the last available version of SILVA database.

Question #6: Which is the best way to normalize or even samples for comparison analysis?

The problems:

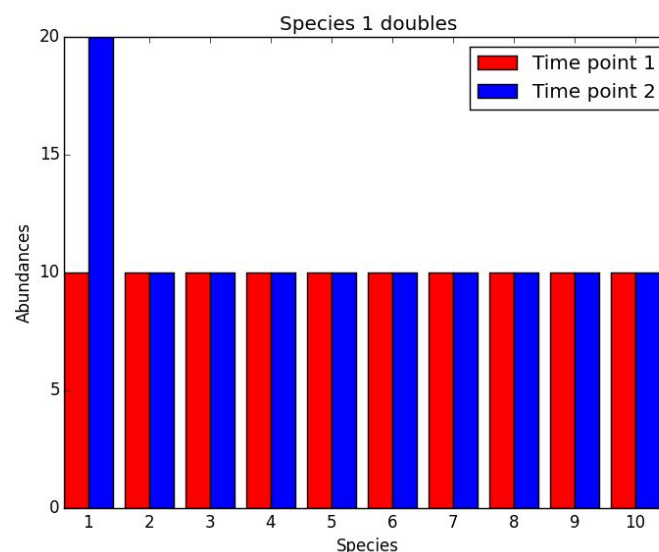
1) Sample size: Microbial communities in each biological sample may be represented by very different numbers of sequences due to differential efficiency of the sequencing process rather than true biological variation. This problem is exacerbated by the observation that the full range of species is rarely saturated. Thus, samples with relatively few sequences may have inflated beta diversity, since authentically shared OTUs are erroneously scored as unique to samples with more sequences.

2) Sparsity: Most OTU tables are sparse, meaning that they contain a high proportion of zero counts (~90%). This implies that the counts of rare OTUs are uncertain, since they are at the limit of sequencing detection ability when there are many sequences per sample and are undetectable when there are few sequences per sample.

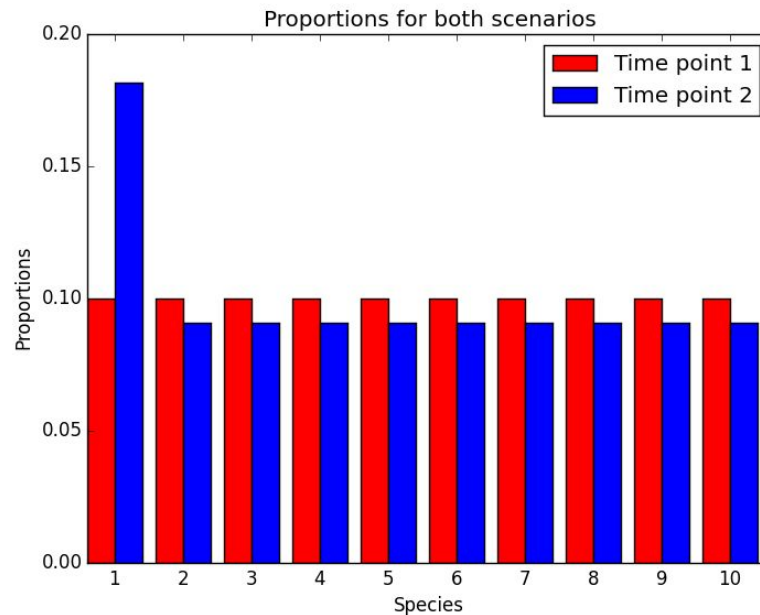
3) Compositional data: It is not possible to obtain the absolute number of microbes from sequence data alone [50]. This is related with the fixed capacity of the sequencing instrument (i.e. it can deliver reads only up to its capacity). Then, the assumption of true independence in the abundance of OTUs or species cannot be held. The total read count observed in a sequencing run is a fixed-size, random sample of the relative abundance of the molecules in the underlying ecosystem. Moreover, the count can not be related to the absolute number of molecules in the input.

Then, some correction/normalization must be made for different samples having different numbers of sequences in order to be comparable. One way to “solve” this is to calculate proportional or relative abundances. This way data gets constrained by the simplex (i.e. sum to 1) and are not free floating in the Euclidean space; for which standard methods of analysis (mainly multivariate) are not applicable. Data that are naturally described as proportions or probabilities, or with a constant sum, are referred to as compositional data. Compositional data contains information about the relationships between the parts. The relationship between absolute abundance in the environment and the relative abundance after sequencing is not predictable. Actually, many people are developing methods to tackle this problem which has been classified as “impossible” [51]. Take a look at [this simple](#) explanation by Jamie Morton:

“A change of 1 species between samples can be also explained by the change of all of the other species between samples. Let's take a look at simple, concrete example.



Here we can observe ten species, and species 1 doubles after the first time point. If we know the original abundances of this species, it's pretty clear that species 1 doubled. However, if we can only obtain the proportions of species within the environment, the message isn't so clear.



Above are the proportions of the species in the exact same environment across the two time points. Here, it looks as everything is changing, which isn't the case in the original environment. Also, there are multiple hypotheses that could explain the change of proportions. For instance, it is a valid hypothesis that species 2-10 all halved. In fact, there are an infinite number of hypotheses that explain this change of proportions, making the differential abundance problem impossibly difficult.”

Answer:

Taking all this into account the answer to our question don't seem to be straightforward, and in fact it is not.

Microbial ecologists commonly normalize their OTU tables by **rarefying** (i.e. drawing without replacement) each sample such that all samples have the same number of total counts. This process, in effect, standardizes the library size across samples, mitigating problem 1) discussed above. Samples with total counts below the defined threshold are excluded, sometimes leading researchers to face difficult trade-offs to choose between sampling depth and the number of samples evaluated.

In this cases, to ensure an informative rarefaction depth is chosen, rarefaction curves can be constructed. These curves plot the number of counts sampled (rarefaction depth) vs. the observed value of species diversity (richness). Rarefaction curves provide a guide that allows users to avoid gutting the species diversity found in samples by choosing a rarefaction depth that is too low.

While rarefying is not an ideal normalization method, as it potentially reduces statistical power depending upon how much data is removed and does not solve the challenge of compositional data, alternatives to rarefying have not been sufficiently developed until recently, and yet complete solutions have not been found [51].

Another common normalization method is **scaling**. Scaling refers to multiplying the matrix counts (OTUs counts) by fixed values or proportions. While microbiome data are frequently sparse as discussed above (in 3), scaling can overestimate or underestimate the prevalence of zero fractions, depending on whether zeros are left in or thrown out of the scaling. This is because putting all samples of varying sampling depth on the same scale ignores the differences in sequencing depth. For example, a rare species having zero counts in a small library size sample can have fractional abundance in a large library size sample. Scaling by the total sum of OTUs in each sample is the most common scaling procedure and leads to tables of **relative abundances** of OTUs and to reinforce the problem 3) of compositional data.

To overcome the compositional data problem some new methods have been proposed. First and simplest, a log-ratio transformation may help. It has been proposed to transform each taxon within a sample by taking the log-ratio of the counts for that taxon divided by the geometric mean of the counts of all taxa, called centred log-ratio [52]. Although this transformation has mathematical elegance, it has potential problems when applied to 16s rRNA based data sets. This difficulty arises from problem 1) and 2): extreme variability of library sizes and great sparsity. In a highly sparse data set, the geometric mean of all taxa can often be zero or near zero. Obviously, if it is zero, a transformation that involves dividing by the geometric mean is undefined. The use of a pseudo-count has been proposed but it is not clear what the value of this pseudo-count should be. Other value may be used for normalizing counts other than the geometric mean. As for RNA-seq data, values based on medians or certain percentiles in the denominator can be used [53]. This offers some of the advantages of the geometric mean, but there is still no guarantee that even very high percentiles do not yield zeros subject to the routinely encountered sparsity. The problems of sparsity also poses general numerical challenges for many traditional tools of statistical analysis. Parametric tests must make accurate estimates of variance for meaningful inference and such estimates are essentially impossible on samples that consist mostly of zeros and no efficient solution has been so far found [51]. The use of nonparametric methods can be a simple solution since they are generally insensitive to factors as pseudocounts addition and avoid making variance estimates that can be skewed by sparse samples. However in 16s rRNA amplicon based experiments there are usually many taxa but few samples and nonparametric methods will lack power to perform inference on differential abundances.

In order to shed some light over this, Weiss et al. (2017) performed a very comprehensive comparison of normalization and transformation methods in association with differential abundance detection strategies using simulated and environmental samples [54]. They tested the effect of rarefaction and scaling by the total sum of OTUs in a column (relative abundances) on the ability to cluster samples by different environments. They also evaluated the effect of this normalization methods on differential abundance (DA) analysis using the Wilcoxon non-parametric test. They also compared this with other methods for differential abundance analysis which include their own normalization methods (i.e. DESeq, edgeR, metagenomeSeq and ANCOM; see Table 3).

They found out that with normalization techniques other than rarefaction, library size is still a frequent confounding factor, especially when using presence/absence data.

Moreover, they found better results when rarefying and using unweighted distances. For weighted distances measures, total sum scaling returns similar results as rarefaction, if the confounding due to sample size is not high.

They showed that rarefying lowers sensitivity (high false negatives) when analyzing differential abundances, even more if combined with non parametric tests. The severity of the decrease in power depends upon how much data has been thrown away and how many samples per group you have. This is why rarefying at the maximum possible depth is recommended. Taking this into account total sum scaling seems more promising , however FDR (false positives) have been shown to increase when differences between samples sizes is over 10X.

The later leads to the idea that is not only a question about how to normalize data but how to choose the best possible combination of normalization and differential abundance analysis method. The mentioned authors have compared the performance of the combinations in Table 3.

Method	Description
Wilcoxon rank-sum test	Also called the Mann-Whitney U test. A non-parametric rank test, which is used on the un-normalized ("None"), proportion normalized, and rarefied matrices
DESeq	nbinom Test—a negative binomial model conditioned test. More conservative shrinkage estimates compared to DESeq2, resulting in stricter type I error control
DESeq2	nbinomWald Test—The negative binomial GLM is used to obtain maximum likelihood estimates for an OTU's log-fold change between two conditions. Then Bayesian shrinkage, using a zero-centered normal distribution as a prior, is used to shrink the log-fold change towards zero for those OTUs of lower mean count and/or with higher dispersion in their count distribution. These shrunken log fold changes are then used with the Wald test for significance
edgeR	exact Test—The same normalization method (in R , method = RLE) as DESeq is utilized, and for differential abundance testing also assumes the NB model. The main difference is in the estimation of the dispersion, or variance, term. DESeq estimates a higher variance than edgeR, making it more conservative in calling differentially expressed OTUs
Voom	Variance modeling at the observational level—library sizes are scaled using the edgeR log counts per million (cpm) normalization factors. Then LOWESS (locally weighted regression) is applied to incorporate the mean-variance trend into precision weights for each OTU
metagenomeSeq	fitZIG—a zero-inflated Gaussian (ZIG) where the count distribution is modeled as a mixture of two distributions: a point mass at zero and a normal distribution. Since OTUs are usually sparse, the zero counts are modeled with the former, and the rest of the log transformed counts are modeled as the latter distribution. The parameters for the mixture model are estimated with an expectation-maximization algorithm, which is coupled with a moderated t statistic fitFeatureModel—a feature-specific zero-inflated lognormal model with empirical Bayes shrinkage of parameter estimates
ANCOM	Analysis of composition of microbiomes—compares the log ratio of the abundance of each taxon to the abundance of all the remaining taxa one at a time. The Mann-Whitney U is then calculated on each log ratio

Table 3: Methods for differential abundance detection benchmarked in Weiss et al. (2017)[54]

Non-parametric tests, as Kruskal Wallis or Mann Whitney, have the advantage of not assumption on the normal distribution of data, which is usually the case with OTU counts and proportions. The main problems of these methods is that they don't account for compositional data and that they are not powerful when sample size is small and sparsity large.

DESeq, DESeq2, Voom and edgeR are RNA-seq derived methods. Because the problem of differential expression in RNA-seq data faces many of the same analysis challenges, these methods have been proposed to be useful for DA analysis. They are based on the negative binomial distribution and attempt to model overdispersion (i.e. the

variances of count distributions are greater than their means) [55]. In particular, metagenomeSeq [56] and ANCOM [57] have been designed for microbial diversity data. metagenomeSeq uses cumulative-sum scaling with the percentile limit determined using a data-driven approach and a zero-inflated Gaussian distribution in order to account for undersampling and sparsity. ANCOM is a novel hypothesis test that accounts for compositional data problems and makes no distributional assumptions.

Weiss et al. results showed that, as expected, rarefaction do not cause a high rate of false discovery but leads to false negatives (lower sensitivity). They conclude that the most promising combinations are: DESeq2, rarefaction + Mann-Whitney and ANCOM.

As regards DESeq2 they confirmed it has high sensitivity for small datasets (<20 samples/group), and also confirmed high false discovery rates [58]. Moreover, adding a pseudocount (e.g. +1) prior DESeq transformations (i.e. log2 and variance stabilizing transformation) increases even more the FDR. ANCOM, which includes Mann-Whitney test, maintained low FDR.

Not yet benchmarked in comparison with all mentioned methods, are novel CoDa (i.e. compositional data) methods [50][51]. Following Aitchison's theory [59], the starting point for any compositional analyses is a ratio transformation of the data. Ratio transformations capture the relationships between the features in the dataset and these ratios are the same whether the data are counts or proportions. Taking the logarithm of these ratios, thus log-ratios, makes the data symmetric and linearly related. Often the centered log-ratio (clr) transformation is used. The clr cannot be determined for sparse data without deleting, replacing or estimating the 0 count values. There are acceptable methods of dealing with 0 count values as both point estimates using zCompositionsR package [60] and as a probability distribution using ALDEx2 available on Bioconductor [61]. Then, ALDEx2 performs statistical tests on the clr values from a modelled probability distribution of the dataset and reports the expected values of parametric and non-parametric statistical tests along with effect-size estimates.

Taking this into account, our recommendations in the UBI are:

- Perform DA analysis on taxonomic category counts and not on OTU counts in order to decrease sparsity.
- Use rarefied data for alpha and beta-diversity analysis.
- Use non-parametric tests (Mann-Whitney and Kruskal Wallis) on data transformed to proportions (total sum scale) to evaluate DA between taxonomic groups in a first step.
- If no differences are found the chosen method may not have enough power and methods for compositional data should be used. ANCOM is the most promising tool, however p-values have not been implemented yet which makes its use not appealing.
- If not using an ASV method, we also recommend to filter the OTU table if DA is to be ran directly over it. OTUs with less than 5 reads should be removed. More strict filters can also be applied.

References:

1. Morgan XC, Huttenhower C. Chapter 12: Human microbiome analysis. PLoS Comput Biol. 2012;8: e1002808.
2. Brooks JP, Edwards DJ, Harwich MD Jr, Rivera MC, Fettweis JM, Serrano MG, et al. The truth about metagenomics: quantifying and counteracting bias in 16S rRNA studies. BMC Microbiol. 2015;15: 66.
3. Gohl DM. The ecological landscape of microbiome science. Nat Biotechnol. 2017;35: 1047–1049.
4. Salter SJ, Cox MJ, Turek EM, Calus ST, Cookson WO, Moffatt MF, et al. Reagent and laboratory contamination can critically impact sequence-based microbiome analyses. BMC Biol. 2014;12: 87.
5. Glassing A, Dowd SE, Galandiuk S, Davis B, Chiodini RJ. Inherent bacterial DNA contamination of extraction and sequencing reagents may affect interpretation of microbiota in low bacterial biomass samples. Gut Pathog. 2016;8: 24.
6. Yeh Y-C, Needham D, Sieradzki E, Fuhrman J. Taxon disappearance from microbiome analysis indicates need for mock communities as a standard in every sequencing run [Internet]. 2017. doi:10.1101/206219
7. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, et al. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol. 2009;75: 7537–7541.
8. Schirmer M, Ijaz UZ, D'Amore R, Hall N, Sloan WT, Quince C. Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. Nucleic Acids Res. 2015;43: e37.
9. Tourlousse DM, Yoshiike S, Ohashi A, Matsukura S, Noda N, Sekiguchi Y. Synthetic spike-in standards for high-throughput 16S rRNA gene amplicon sequencing. Nucleic Acids Res. 2017;45: e23.
10. Davis NM, Proctor D, Holmes SP, Relman DA, Callahan BJ. Simple statistical identification and removal of contaminant sequences in marker-gene and metagenomics data [Internet]. 2017. doi:10.1101/221499
11. Dohm JC, Lottaz C, Borodina T, Himmelbauer H. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. Nucleic Acids Res. 2008;36: e105.
12. Bokulich NA, Subramanian S, Faith JJ, Gevers D, Gordon JI, Knight R, et al. Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. Nat Methods. 2013;10: 57–59.
13. Dickie IA. Insidious effects of sequencing errors on perceived diversity in molecular surveys. New Phytol. 2010;188: 916–918.

14. Krohn A, Stevens B, Robbins-Pianka A, Belus M, Allan GJ, Gehring C. Optimization of 16S amplicon analysis using mock communities: implications for estimating community diversity [Internet]. 2016. doi:10.7287/peerj.preprints.2196v1
15. Nguyen NH, Smith D, Peay K, Kennedy P. Parsing ecological signal from noise in next generation amplicon sequencing. *New Phytol.* 2015;205: 1389–1393.
16. [No title] [Internet]. [cited 23 Aug 2018]. Available: https://www.researchgate.net/profile/Samuel_Viana/publication/312491639_Optimizing_16S_Analysis_Pipelines_thesis_final_version/links/58add0ce45851503be91e4d4/Optimizing-16S-Analysis-Pipelines-thesis-final-version.pdf
17. Kozich JJ, Westcott SL, Baxter NT, Highlander SK, Schloss PD. Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq Illumina sequencing platform. *Appl Environ Microbiol.* 2013;79: 5112–5120.
18. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. DADA2: High-resolution sample inference from Illumina amplicon data. *Nat Methods.* 2016;13: 581–583.
19. Mysara M, Leys N, Raes J, Monsieurs P. IPED: a highly efficient denoising tool for Illumina MiSeq Paired-end 16S rRNA gene amplicon sequencing data. *BMC Bioinformatics.* 2016;17: 192.
20. Aronesty E. Comparison of Sequencing Utility Programs. *Open Bioinforma J.* 2013;7: 1–8.
21. Callahan BJ, McMurdie PJ, Holmes SP. Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. *ISME J.* 2017;11: 2639–2643.
22. Schloss PD, Handelsman J. Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. *Appl Environ Microbiol.* 2005;71: 1501–1506.
23. Edgar RC. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics.* 2010;26: 2460–2461.
24. Albanese D, Fontana P, De Filippo C, Cavalieri D, Donati C. MICCA: a complete and accurate software for taxonomic profiling of metagenomic data. *Sci Rep.* 2015;5. doi:10.1038/srep09743
25. Mahé F, Rognes T, Quince C, de Vargas C, Dunthorn M. Swarm: robust and fast clustering method for amplicon-based studies [Internet]. 2014. doi:10.7287/peerj.preprints.386
26. Schloss PD, Westcott SL. Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Appl Environ Microbiol.* 2011;77: 3219–3226.

27. Rideout JR, He Y, Navas-Molina JA, Walters WA, Ursell LK, Gibbons SM, et al. Subsampled open-reference clustering creates consistent, comprehensive OTU definitions and scales to billions of sequences. *PeerJ*. 2014;2: e545.
28. Amir A, McDonald D, Navas-Molina JA, Kopylova E, Morton JT, Zech Xu Z, et al. Deblur Rapidly Resolves Single-Nucleotide Community Sequence Patterns. *mSystems*. 2017;2. doi:10.1128/mSystems.00191-16
29. Edgar RC. UNOISE2: improved error-correction for Illumina 16S and ITS amplicon sequencing [Internet]. 2016. doi:10.1101/081257
30. Huang W, Li L, Myers JR, Marth GT. ART: a next-generation sequencing read simulator. *Bioinformatics*. 2011;28: 593–594.
31. Kopylova E, Navas-Molina JA, Mercier C, Xu ZZ, Mahé F, He Y, et al. Open-Source Sequence Clustering Methods Improve the State Of the Art. *mSystems*. 2016;1. doi:10.1128/mSystems.00003-15
32. Edgar RC. Updating the 97% identity threshold for 16S ribosomal RNA OTUs [Internet]. 2017. doi:10.1101/192211
33. Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Ostell J, Pruitt KD, et al. GenBank. *Nucleic Acids Res*. 2018;46: D41–D47.
34. Wang Q, Garrity GM, Tiedje JM, Cole JR. Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Appl Environ Microbiol*. 2007;73: 5261–5267.
35. Allard G, Ryan FJ, Jeffery IB, Claesson MJ. SPINGO: a rapid species-classifier for microbial amplicon sequences. *BMC Bioinformatics*. 2015;16: 324.
36. Liu Z, DeSantis TZ, Andersen GL, Knight R. Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers. *Nucleic Acids Res*. 2008;36: e120.
37. Yilmaz P, Parfrey LW, Yarza P, Gerken J, Priesse E, Quast C, et al. The SILVA and “All-species Living Tree Project (LTP)” taxonomic frameworks. *Nucleic Acids Res*. 2013;42: D643–D648.
38. McDonald D, Price MN, Goodrich J, Nawrocki EP, DeSantis TZ, Probst A, et al. An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J*. 2012;6: 610–618.
39. Edgar R. SINTAX: a simple non-Bayesian taxonomy classifier for 16S and ITS sequences [Internet]. 2016. doi:10.1101/074161
40. Maidak BL, Cole JR, Lilburn TG, Parker CT Jr, Saxman PR, Farris RJ, et al. The RDP-II (Ribosomal Database Project). *Nucleic Acids Res*. 2001;29: 173–174.
41. Priesse E, Quast C, Knittel K, Fuchs BM, Ludwig W, Peplies J, et al. SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA

- sequence data compatible with ARB. *Nucleic Acids Res.* 2007;35: 7188–7196.
42. Yarza P, Yilmaz P, Pruesse E, Glöckner FO, Ludwig W, Schleifer K-H, et al. Uniting the classification of cultured and uncultured bacteria and archaea using 16S rRNA gene sequences. *Nat Rev Microbiol.* 2014;12: 635–645.
 43. DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, Keller K, et al. Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl Environ Microbiol.* 2006;72: 5069–5072.
 44. Balvočiūtė M, Huson DH. SILVA, RDP, Greengenes, NCBI and OTT — how do these taxonomies compare? *BMC Genomics.* 2017;18. doi:10.1186/s12864-017-3501-4
 45. Karsch-Mizrachi I, Takagi T, Cochrane G, International Nucleotide Sequence Database Collaboration. The international nucleotide sequence database collaboration. *Nucleic Acids Res.* 2018;46: D48–D51.
 46. Parte AC. LPSN—list of prokaryotic names with standing in nomenclature. *Nucleic Acids Res.* 2013;42: D613–D616.
 47. Price MN, Dehal PS, Arkin AP. FastTree: computing large minimum evolution trees with profiles instead of a distance matrix. *Mol Biol Evol.* 2009;26: 1641–1650.
 48. Garrity GM, Holt JG, Castenholz RW, Pierson BK, Keppen OI, Gorlenko VM. Phylum BVI. Chloroflexi phy. nov. *Bergey's Manual® of Systematic Bacteriology.* 2001. pp. 427–446.
 49. Almeida A, Mitchell AL, Tarkowska A, Finn RD. Benchmarking taxonomic assignments based on 16S rRNA gene profiling of the microbiota from commonly sampled environments. *Gigascience.* 2018;7. doi:10.1093/gigascience/giy054
 50. Gloor GB, Macklaim JM, Pawlowsky-Glahn V, Egozcue JJ. Microbiome Datasets Are Compositional: And This Is Not Optional. *Front Microbiol.* 2017;8. doi:10.3389/fmicb.2017.02224
 51. Tsilimigras MCB, Fodor AA. Compositional data analysis of the microbiome: fundamentals, tools, and challenges. *Ann Epidemiol.* 2016;26: 330–335.
 52. Fernandes AD, Reid JN, Macklaim JM, McMurrough TA, Edgell DR, Gloor GB. Unifying the analysis of high-throughput sequencing datasets: characterizing RNA-seq, 16S rRNA gene sequencing and selective growth experiments by compositional data analysis. *Microbiome.* 2014;2: 15.
 53. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2 [Internet]. 2014. doi:10.1101/002832
 54. Weiss S, Xu ZZ, Peddada S, Amir A, Bittinger K, Gonzalez A, et al. Normalization and microbial differential abundance strategies depend upon data characteristics. *Microbiome.* 2017;5: 27.
 55. Dillies M-A, Rau A, Aubert J, Hennequet-Antier C, Jeanmougin M, Servant N, et al. A

- comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief Bioinform.* 2013;14: 671–683.
56. Paulson JN, Stine OC, Bravo HC, Pop M. Differential abundance analysis for microbial marker-gene surveys. *Nat Methods.* 2013;10: 1200–1202.
 57. Mandal S, Van Treuren W, White RA, Eggesbø M, Knight R, Peddada SD. Analysis of composition of microbiomes: a novel method for studying microbial composition. *Microb Ecol Health Dis.* 2015;26. doi:10.3402/mehd.v26.27663
 58. Hawinkel S, Mattiello F, Bijmans L, Thas O. A broken promise: microbiome differential abundance methods do not control the false discovery rate. *Brief Bioinform.* 2017; doi:10.1093/bib/bbx104
 59. Aitchison J. *The Statistical Analysis of Compositional Data.* 1986.
 60. Palarea-Albaladejo J, Martín-Fernández JA. zCompositions — R package for multivariate imputation of left-censored data under a compositional approach. *Chemometrics Intellig Lab Syst.* 2015;143: 85–96.
 61. Bian G, Gloor GB, Gong A, Jia C, Zhang W, Hu J, et al. The Gut Microbiota of Healthy Aged Chinese Is Similar to That of the Healthy Young. *mSphere.* 2017;2. doi:10.1128/mSphere.00327-17
 62. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, et al. QIIME allows analysis of high-throughput community sequencing data. *Nat Methods.* 2010;7: 335–336.

Acknowledgement:

This work was supported by **ONEIDA project** (LISBOA-01-0145-FEDER-016417) co-funded by FEEI - "Fundos Europeus Estruturais e de Investimento" from "Programa Operacional Regional Lisboa 2020" and by national funds from FCT - "Fundação para a Ciência e a Tecnologia."

