

A novel conceptual approach to read-filtering in high-throughput amplicon sequencing studies

Fernando Puente-Sánchez^{1,*}, Jacobo Aguirre^{1,2,3} and Víctor Parro¹

¹Department of Molecular Evolution, Centro de Astrobiología (INTA-CSIC). Instituto Nacional de Técnica Aeroespacial, Ctra de Torrejón a Ajalvir km 4. 28850 Torrejón de Ardoz, Madrid, Spain, ²Centro Nacional de Biotecnología (CSIC). c/ Darwin 3, 28049 Madrid, Spain and ³Grupo Interdisciplinar de Sistemas Complejos (GISC), Madrid, Spain

Received April 20, 2015; Revised September 08, 2015; Accepted October 12, 2015

ABSTRACT

Adequate read filtering is critical when processing high-throughput data in marker-gene-based studies. Sequencing errors can cause the mis-clustering of otherwise similar reads, artificially increasing the number of retrieved Operational Taxonomic Units (OTUs) and therefore leading to the overestimation of microbial diversity. Sequencing errors will also result in OTUs that are not accurate reconstructions of the original biological sequences. Herein we present the Poisson binomial filtering algorithm (PBF), which minimizes both problems by calculating the error-probability distribution of a sequence from its quality scores. In order to validate our method, we quality-filtered 37 publicly available datasets obtained by sequencing mock and environmental microbial communities with the Roche 454, Illumina MiSeq and IonTorrent PGM platforms, and compared our results to those obtained with previous approaches such as the ones included in mothur, QIIME and USEARCH. Our algorithm retained substantially more reads than its predecessors, while resulting in fewer and more accurate OTUs. This improved sensitiveness produced more faithful representations, both quantitatively and qualitatively, of the true microbial diversity present in the studied samples. Furthermore, the method introduced in this work is computationally inexpensive and can be readily applied in conjunction with any existent analysis pipeline.

INTRODUCTION

High-throughput sequencing of marker genes, such as the 16S ribosomal RNA, has become an invaluable tool for microbial ecologists, since it allows for a previously unreachable level of detail in the analysis of complex microbial communities. Many studies have used platforms such as

the Roche 454, Illumina or IonTorrent sequencers to thoroughly characterize and compare microbial communities at an affordable cost (1–5), while others have taken advantage of their very high yield in order to analyze the structure and composition of the rare biosphere (6). However, the correct assessment of sequencing artifacts is critical in obtaining representative results. Reads derived from the same biological template may differ due to sequencing errors, which can cause them to be assigned to different clusters and therefore result in the overestimation of microbial diversity (7). The most common software tools and packages include sequence clustering into OTUs in their recommended pipelines (8–15, see (15) for a comparison of several molecular ecology pipelines). Alternatives to traditional clustering have been recently proposed, such as distribution-based clustering (16) or a clustering-free approach (17). These novel methods are specially suited for subpopulation level studies, but work only for moderate-to-high abundance sequences, being unsuitable for population-level alpha or beta diversity studies (17). Moreover, even although they can remove likely erroneous sequences and resolve subpopulations based on dynamic information, they nevertheless rely on a quality filtering step for the preprocessing of raw reads (17).

Amplicon denoising (18,19) is a widespread method for filtering Roche 454 pyrosequencing reads that can also be applied to IonTorrent data. It works on flowgrams rather than sequences, which allows for a more natural modeling of the homopolymer read errors that are characteristic of pyrosequencing and ion semiconductor sequencing. However, it is platform specific and computationally expensive.

For Illumina systems, there is no consensus approach to quality filtering, with the authors of mothur (20), QIIME (21) and UPARSE (15) proposing different solutions. All those heuristic approaches were published as parts of their respective pipelines, but to the best of our knowledge they have not been thoroughly compared to each other.

The lack of a rigorous method for incorporating quality scores in the analysis of marker-gene sequences has also led some authors to advocate for a stringent filtering in or-

*To whom correspondence should be addressed. Email: fpusan@gmail.com

der to reduce the retrieval of spurious diversity (3). However, overstringent filtration will result in an undesired loss of sensitivity and will have an impact on the observed taxonomic distribution (21). Therefore, an accurate algorithm that overcomes these problems is desirable.

Herein we present and validate the *Poisson binomial filtering* (PBF) method, which is able to determine the error-probability distribution of any sequence with associated quality scores, by using a simple statistical approach. Phred quality scores, which represent the probability that a given base call is mistaken, can be derived from the raw output of every sequencing platform. Reading a single base can be likened to tossing a coin: the base is either right or wrong, and both chances can be determined from its quality score. In fact, the number of errors present in a given base follows a Bernoulli distribution, i.e. a binomial distribution with a single trial. For a sequence of nucleotides with potentially non-equal error probabilities, we sum their associated Bernoulli random variables in order to obtain the estimated probability that the sequence has accumulated more than k errors, where k is the maximum number of errors that still allows for a correct clustering (Supplementary Note 1). When compared with the filtering approaches included in mainstream molecular ecology pipelines such as mothur, QIIME or UPARSE, Poisson binomial filtering proved to be the most accurate algorithm for filtering marker-gene sequences. Additionally, PBF is based on simple statistical principles and, since it only requires Phred quality scores as an input, it is expected to work robustly regardless of the sequencing platform, assuming that the quality scores are acceptable predictors of the true error probabilities, and that errors are independent. Finally, our algorithm is computationally efficient, scales linearly with the number of sequences, and has a low memory fingerprint, making it useful even in low-performance desktop environments.

MATERIALS AND METHODS

The Poisson binomial filtering algorithm

Let us suppose we have 1 sequence of length N nucleotides (nt), each nucleotide with a potentially non-equal probability p_i of being erroneous and a probability $(1-p_i)$ of being correct. Our target is to obtain the probability of this sequence of having j erroneous nucleotides, for $j = 0, 1, 2, \dots, N$ (see example in Figure 1a,b). Statistically, our problem can be analyzed as the probability distribution of the number of successes in a sequence of N independent yes/no experiments with success probabilities p_1, p_2, \dots, p_N . This is equivalent to the sum S_N of N independent Bernoulli distributed random variables X_1, X_2, \dots, X_N such that $S_N = \sum_{i=1}^N X_i$, where

$$\begin{aligned} P(X_i = j) &= 1 - p_i \text{ for } j = 0, \\ P(X_i = j) &= p_i \text{ for } j = 1, \\ P(X_i = j) &= 0 \text{ for } j > 1, \end{aligned} \quad (1)$$

and $P(X_i = j)$ stands for the probability of obtaining j errors in nucleotide i . The stochastic variable S_N follows a Poisson binomial distribution (Supplementary Note 1), from where we name the method presented here.

While the probability of obtaining a sequence with j errors in a sequence, for all values of j , can be expressed explicitly (see Eq. (SN1.2) and its derivation in section Supplementary Note SN1.1), it becomes useless in practice for moderate values of j . We explain here an alternative algorithm inspired by (22) that allows us to calculate the error-probability distribution $P(S_N = j)$ for all j in a simple and efficient way.

First, note that if we have two random variables Y and Z , each of them taking discrete values $0, 1, 2, \dots$, the probability of the sum $Y+Z$ of taking value j is

$$P(Y + Z = j) = \sum_{i=0}^j P(Y = i)P(Z = j - i). \quad (2)$$

The algorithm results:

1. Obtain $P(X_1 = j)$ from Equation (1). Let $U = X_1$.
2. For $i = 2, 3, \dots, N$, the distribution is obtained by following (a–c) recursively.
 - (a) Calculate $P(X_i = j)$ from Equation (1).
 - (b) Calculate $P(Y+Z = j)$ from Equation (2), being $Y = U$ and $Z = X_i$.
 - (c) Let $U = Y+Z$.
3. The estimated probability for the sequence under study of having j errors, $P(S_N = j)$, is given by U when $i = N$.
4. The steps (1–3) must be repeated for $j = 0, 1, 2, \dots, j_{\max}$, where j_{\max} is the lowest value of j that satisfies $\sum_{r=0}^j P(S_N = r) \geq \xi$ and $0 < \xi < 1$ is a confidence coefficient (in our case $\xi = 0.995$). Let j_ξ be the number such that the sequence has a probability ξ of having less than j_ξ errors. It is obtained interpolating the accumulated error probability of the sequence between the values $r = j_{\max} - 1$ and $r = j_{\max}$ to obtain its exact value in $r = j_\xi$. A linear interpolation yields

$$j_\xi = j_{\max} - 1 + \frac{\xi - \sum_{r=0}^{j_{\max}-1} P(S_N = r)}{P(S_N = j_{\max})}.$$

5. Let j_{tol} be the maximum tolerable number of errors per sequence, that is, the maximum number of errors allowed for a correct clustering. In our calculations, we have fixed j_{tol} at 1% of the trimmed sequence length (that is, at 2.5 for the 454 and Illumina datasets, and at 2 for the Ion-Torrent datasets). The sequence under study is discarded if $j_\xi > j_{\text{tol}}$, and accepted otherwise (Figure 1b,c). At this moment, the calculation for this particular sequence is finished, and it is time to repeat the whole algorithm for the rest of the sequences of the population.

j_ξ is therefore the predicted maximum number of errors of a given sequence, with a confidence coefficient of ξ . Under the parameters used in this study, a sequence has an *alpha* = $(1 - \xi) = 0.005$ probability of having more than j_ξ errors (assuming that Q scores accurately reflect true error probabilities, see discussion for more details). In practice, this also means that j_ξ will be an overestimation of the actual number of errors. This can be easily avoided by the users by setting a different confidence coefficient. However, we also want to

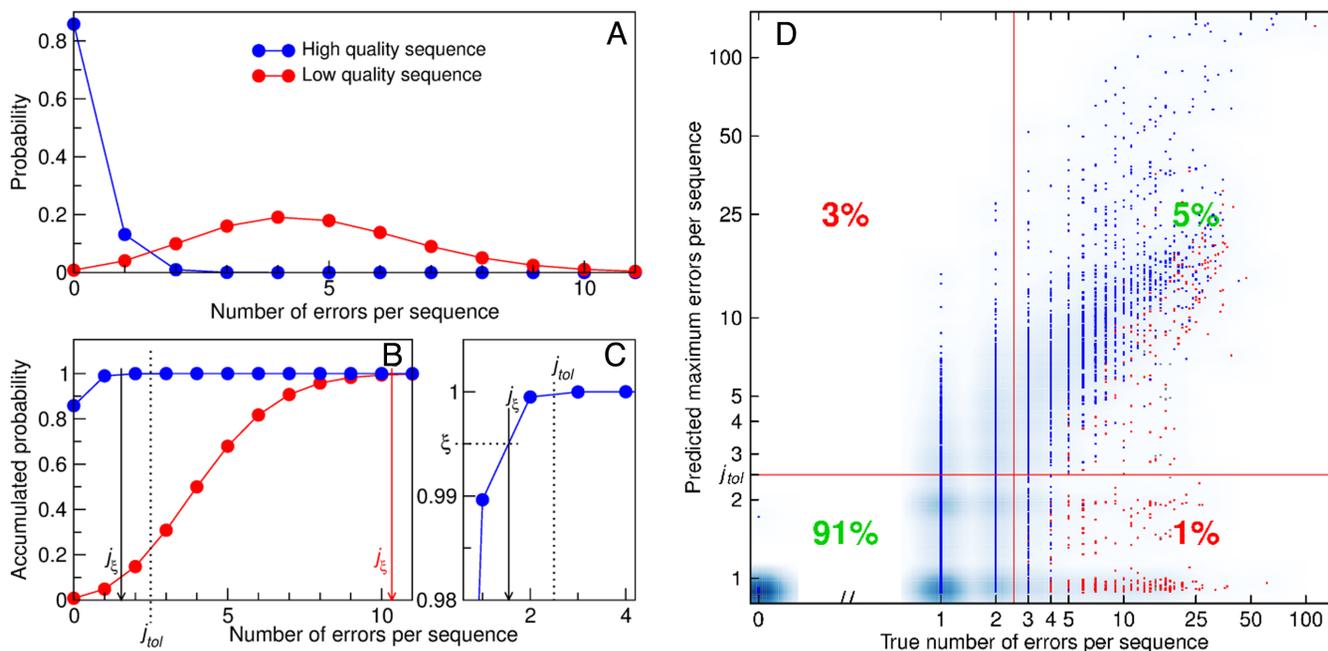


Figure 1. Poisson binomial filtering accurately discriminates between good and erroneous sequences. (A, B): Error probability distribution (A) and accumulated error probability distribution (B) of two example nucleotide sequences, as calculated from their quality scores by the Poisson binomial filtering algorithm. j_{ξ} stands for the 99.5th percentile of the error probability (i.e. a sequence has a probability $\xi = 0.995$ of having less than j_{ξ} errors). j_{tol} is the maximum tolerable number of errors (1% of the sequence length in our case). Sequences with $j_{\xi} > j_{tol}$ are discarded in the filtering step. (C) Zoom of (B), sketching the calculation of j_{ξ} for the high quality sequence. (D) Comparison between the predicted maximum errors j_{ξ} calculated by the Poisson binomial algorithm and the true number of errors for all sequences from the EvenIT mock community dataset. Dots represent unique sequences. True mock community sequences are plotted in blue, contaminant sequences are plotted in gray, and chimeric sequences are plotted in red. The blue background represents sequence abundance (note that few unique sequences may have a high number of representatives, and vice versa). Red lines indicate our error cut-off of 2.5 errors per sequence (j_{tol}). The plot is thus divided in four quadrants corresponding to correctly retained sequences (lower left), correctly discarded sequences (upper right), incorrectly discarded sequences (upper left) and incorrectly retained sequences (lower right). The percentage of true mock community sequences present on each quadrant is also indicated. PBF correctly classified 96% of the non-chimeric/non-contaminant sequences present in the EvenIT dataset. The graph is plotted in logarithmic scale (the 0 in the x-axis is added for clarity).

note that our recommended parameters provide a clear upper bound to the number of errors which is unlikely to be trespassed, and nonetheless result in a very high proportion of correctly classified sequences (see results and Figure 1d).

Finally, as our problem corresponds to the sum of N binomial distributions of probabilities p_i and number of trials $n = 1$, it can be approximated to a Poisson distribution as far as N is high and $p_i \ll 1$. The Poisson approximated probability for the sequence under study of having j errors, $P(S_N = j)$, becomes

$$P(S_N = j) = \frac{\lambda^j \exp(-\lambda)}{j!},$$

where

$$\lambda = \sum_{i=1}^N p_i.$$

While this approximation is reasonably accurate and quicker to compute, it may fail even for high-quality sequences, provided they contain one or more low-quality bases. This occurrence is not uncommon in real datasets, and can lead to significant differences between the Poisson binomial filtering algorithm and its Poisson approximation (Supplementary Note SN1.5).

A more detailed explanation of the Poisson binomial filtering algorithm presented above and its Poisson approximation can be found in Supplementary Note 1.

Algorithm implementation

Both C and Python implementations of the Poisson binomial filtering algorithm are available in GitHub (<http://github.com/fpusan/moira>), and as a pip-installable python package (which can be installed by typing ‘pip install moira’ in a linux, mac or windows command line). The execution time of the Poisson binomial algorithm and its Poisson approximation for increasing numbers of sequences and for sequences of increasing length can be found in Supplementary Note 8. Full documentation is available at <https://github.com/fpusan/moira/blob/master/README.md>. The Poisson binomial filtering algorithm has also been included in the LotuS OTU processing pipeline (23, <http://psbweb05.psb.ugent.be/lotus/>).

The moira filtering pipeline

The script *moira.py* contains an implementation of the Poisson binomial filtering algorithm and performs the following tasks:

- If required, it assembles contigs from paired reads (*-paired*). The assembler is an implementation of mothur make.contigs command (<http://www.mothur.org/wiki/Make.contigs>), and includes a modified version of the Needleman-Wunsch global aligner and a consensus sequence constructor. Our implementation also returns consensus quality scores, which are simply the highest quality scores for each position of the alignment.
- It truncates sequences to a fixed length (*-truncate*), discarding the sequences that are smaller than the cut-off.
- It applies the Poisson binomial algorithm to calculate the predicted maximum number of errors of each remaining sequence, with a given confidence coefficient (settable by the *-alpha* parameter) and discards the ones that have more errors per nucleotide than the specified cut-off (*-uncert*). The *alpha* parameter is defined as $1 - \xi$, and represents the probability of underestimating the errors present on a given sequence.
- It collapses identical sequences and chooses the one with the least predicted maximum errors as the group representative for filtering (*-collapse*). We assumed that, in spite of differences in quality, identical sequences should have the same origin, as it is unlikely that two biologically unrelated sequences become identical due to sequencing errors. Thus, if one of them has good quality, the rest should be considered as true biological sequences and be allowed into the final dataset. We have demonstrated that collapsing sequences prior to quality filtering actually helps to mitigate an important source of taxonomic bias during sequence processing (Supplementary Note 2).

16S mock community data

Two synthetic mock microbial communities designed by the Human Microbiome Project (24, <http://www.hmpdacc.org/HMMC>) were used for evaluating the different filtering methods. Genetic DNA from 22 different organisms (20 bacterial, 1 archaeal and 1 eukaryotic) was mixed in known amounts, based on qPCR of the small subunit (SSU) rRNA gene, in order to generate two different mixtures: an Even mock community, in which there is a similar amount of SSU rRNA copies for each organism, and a Staggered mock community, in which the amounts of SSU rRNA of each organism are different.

The data used in this study come from publicly available libraries generated by sequencing the Even and Staggered mock communities with the Roche 454 GS FLX Titanium, the Illumina MiSeq and the IonTorrent PGM platforms. References for all the datasets used in this study are given in Supplementary Note 3.

Validation of Poisson binomial filtering on mock community data

The script *moira.py* was used to predict the number of errors present on each sequence for all the six Roche 454 GS FLX Titanium, the four Illumina MiSeq and the two IonTorrent PGM mock community datasets. For the MiSeq datasets, contigs were first assembled from paired-end reads by applying the *-paired* flag. The *-alpha* parameter, which

indicates the probability of a read having more errors than reported, was left as its default value of 0.005. Identical reads were collapsed and the sequence with the smallest number of errors was chosen as the group representative, as described above. These predicted maximum values were compared to the true number of errors of each sequence, which was obtained by using the mothur command *seq.error*. Briefly, the sequences were aligned to a reference database made up from the true biological sequences present in the mock community (which can be found in http://www.mothur.org/wiki/454_SOP). Sequences with less than 80% alignment coverage were discarded at this step. The resulting alignment was then used to determine the true number of errors present on each sequence, as well as whether that sequence was chimeric or not, bearing in mind that chimeras cannot be identified with absolute certainty. Likely non-chimeric sequences that nevertheless showed less than 95% similarity to their best hit in the mock reference database were aligned again against mothur's SILVA 16S reference alignment (25, version 98). In case said sequence showed a pairwise identity and an alignment coverage equal or greater to 95% to any sequence in the 16S reference alignment, it was considered to be a contaminant.

Quality filtering of 16S reads

USEARCH. Trimming of reads by quality values was performed by using the USEARCH *fastq.filter* command, as employed by (15). Reads (for 454/IonTorrent data) or contigs (for paired Illumina data) were truncated at the first position with a quality score below 15 (*-fastq.trunqual* 15). After that, sequences were truncated to a length of 250 nucleotides (200 nucleotides for IonTorrent data), and sequences smaller than 250 nt (200 nt for IonTorrent data) were discarded (*-fastq.trunclen* 250/200).

We also tested a different method implemented in the USEARCH *fastq.filter* command, as suggested in the author's web page (http://drive5.com/usearch/manual/uparse_cmds.html). Briefly, reads (for 454 data and IonTorrent) or contigs (for paired MiSeq data) with more than 0.5 expected errors (*-fastq.maxee* 0.5) were discarded. After that, sequences were truncated to a length of 250 nt (200 nt for IonTorrent data), and sequences smaller than 250 nt (200nt for IonTorrent data) were discarded (*-fastq.trunclen* 250/200). NOTE: During the review process of this manuscript, more details on the USEARCH expected errors method were published in (26). We now discuss the differences and similarities between the USEARCH expected errors filter and our own in Supplementary Note 9.

Mothur. Denoising of 454 and IonTorrent reads was performed using the mothur command *shhh.flows*, which is an implementation of the PyroNoise algorithm, as recommended in the mothur SOP (www.mothur.org/wiki/Analysis_examples). After denoising, mothur command *trim.seqs* was used to truncate the denoised sequences to a length of 250 nt, and to discard sequences smaller than 250 nt (200 nt in both cases for IonTorrent data).

Additionally, paired Illumina reads were assembled and filtered according to mothur's MiSeq SOP (<http://www>.

mothur.org/wiki/MiSeq_SOP). After filtering, mothur command *trim.seqs* was used to truncate the contigs to a length of 250 nt, and to discard sequences smaller than 250 nt.

QIIME. Paired Illumina reads were assembled with QIIME's *join_paired_ends.py* script, using the default parameters. QIIME's script *split_libraries_fastq.py* was used to filter the resulting contigs, as recommended by the authors (*-r 3 -p 0.75 -q 3 -n 0*) in (21).

Poisson binomial filtering. The script *moira.py* was used to perform Poisson binomial filtering on 454/IonTorrent reads or contigs assembled from Illumina paired reads (*-paired*), as described above. Before filtering, sequences or contigs were truncated to 250 nt, and the sequences smaller than 250 nt (200 nt in both cases for IonTorrent reads) were discarded (*-truncate 250/200*). Identical 454/IonTorrent reads or Illumina contigs were clustered together prior to quality control (*-collapse*) and the sequence with the highest quality was chosen as the group representative for quality control. 0.01 or less errors per nucleotide were tolerated (*-uncert 0.01*) with a 0.005 chance of error underestimation (*-alpha 0.005*).

For each method, paired Illumina reads were assembled as recommended by its authors.

Note that, for consistency, we have chosen the 250 nt cut-off recommended by (15) as the fixed length for the rest of the filtering methods, for the 454 and Illumina datasets. Since read length may have an effect in clustering and OTU accuracy, we believe that equalizing it results in more valid comparisons between the different filtering methods. In a similar fashion, the 200 nt cutoff proposed in <http://www.brmicrobiome.org/#!16sprofilingpipeline/cuhd> was applied to all the filtering methods for the IonTorrent datasets.

The full list of commands used for each method can be found in Supplementary Note 10.

Common processing pipeline for the filtered reads

Regardless of the filtering method, the filtered sequences were subjected to a common pipeline based in mothur's recommended SOP that included the following steps:

- Sequence alignment to mothur's SILVA Reference Alignment.
- Optimization of the alignment space by removing the sequences that failed to align correctly, in order to ensure that all the remaining sequences overlap at the same region of the SILVA Reference Alignment.
- Pre-clustering of similar sequences.
- Removal of chimeras with UCHIME.
- Taxonomic classification and removal of non-bacterial and unclassified sequences.
- Library size standardization (see below).
- Clustering of the remaining sequences using mothur's default average neighbor algorithm, with an OTU distance cut-off of 0.03.
- Accuracy classification of the resulting OTUs (see below).

For each sample, the libraries obtained after filtering the raw reads with the different methods were standardized

to a similar size by random sub-sampling. Total number of retrieved OTUs and singleton OTUs, as well as accuracy assessment of the OTU representative sequences, were obtained by averaging the results from 100 independent rounds of random library size standardizations followed by clustering of the resulting reads.

The full list of commands can be found in Supplementary Note 10.

OTU accuracy assessment on mock communities

The accuracy of the obtained OTU representative sequences was evaluated by aligning them to a reference database made up from the true biological sequences present in the sample, as previously described by (15). Sequence alignment was performed with mothur *align.seqs* command. If the pairwise identity of an OTU representative sequence to any sequence in the reference database was 100%, the OTU was classified as 'Perfect'. If the pairwise identity was smaller than 100%, but greater or equal to 99%, the OTU was classified as 'Good'. If the pairwise identity was smaller than 99% but greater or equal to 97%, the OTU was classified as 'Noisy'. If the pairwise identity was lower than 95%, the OTU representative sequence was aligned to mothur's SILVA bacterial 16S reference alignment (version 98). If said sequence showed a pairwise identity and an alignment coverage equal or greater to 95% to any sequence in the 16S reference alignment, the OTU was classified as 'Contaminant'. When none of the above conditions applied, the OTU was considered to be the result of either an undetected chimera, a mock community sequence with more than 3% errors or a novel contaminant sequence with more than 5% errors, and was classified as 'Other'.

OTU accuracy assessment on environmental communities

OTU representative sequences from environmental communities were aligned with mothur's SILVA bacterial 16S reference alignment (25). For each dataset and filtering method, the average similitude of the OTU representative sequences to their best hits in the SILVA alignment was calculated. This was taken as an indicator of the overall accuracy of the resulting OTUs, under the assumption that sequencing errors are more likely to decrease OTU similitude to known sequences than to increase it.

References for all the environmental datasets used in this study are given in Supplementary Note 3.

Assessment of the taxonomic bias caused by the different filtering methods

Taxonomic bias was assessed by comparing the taxonomic composition of the sample before and after performing quality filtering. Sequences were classified by using the *classify.seqs* command implemented in mothur and mothur's RDP 16S rRNA reference database (version 9). Then, taxonomic composition was obtained by calculating the proportion of sequences that were assigned to each phylotype at the genus level with an 80% confidence cut-off (40% for the environmental communities). Finally, taxonomic bias was calculated as the Bray-Curtis dissimilarity between the filtered and unfiltered sequence communities. In the 454 and

IonTorrent libraries from the environmental communities, a high proportion of sequences did not get classified at the genus level. Therefore, the taxonomic composition of those libraries was instead calculated at the class level.

RESULTS

We validated the Poisson binomial filtering algorithm and compared it with the different filtering approaches recommended by the authors of *mothur* (8,13,20), USEARCH-UPARSE (10,15,26) and QIIME (21) by quality-filtering datasets obtained by sequencing different mock and environmental microbial communities with the Roche 454 GS FLX Titanium, the Illumina MiSeq and the IonTorrent PGM platforms. In order to evaluate the different methods on equal grounds, filtered reads were processed with a common downstream pipeline that included chimera-filtering with UCHIME (27), sample size standardization and OTU clustering.

PBF accurately discriminates between good and erroneous sequences

When applying our default cut-off of 1% errors allowed per sequence, our algorithm accurately classified 96% of the mock community sequences from the Even1M dataset (Figure 1d). Three percent of the sequences were incorrectly discarded while, remarkably, only 1% of the sequences were incorrectly retained. Moreover, most of those incorrectly retained sequences had only three true errors (1.2% errors per sequence), meaning that they would likely cluster correctly when applying the standard 3% OTU distance cut-off. The rest of the Illumina datasets rendered similar results. The accuracy of our method was slightly lower for the 454 and IonTorrent datasets, but it nevertheless resulted in a minimum of 88% (for 454) and 79% (for IonTorrent) correctly classified sequences (Supplementary Figure SN4.1).

Performance of the different filtering methods on mock community datasets

Publicly available datasets from even and staggered mock communities from the Human Microbiome Project (24) were filtered with PBF, *mothur*, USEARCH and QIIME (Figure 2, Supplementary Note 4). These artificial communities contain known amounts of 16S rRNA gene copies from 20 different bacterial organisms. The fact that both the qualitative and quantitative composition of the samples are known beforehand allowed us to thoroughly compare the effects of the different filtering methods in terms of OTU accuracy, alpha diversity and community composition. OTU accuracy was defined as the maximum similarity of its representative sequence to the 16S sequences of the microorganisms used to build the mock community, as previously described in (15). We were also interested in determining how the different filtering processes affected the observed community composition. The taxonomic bias in community composition caused by any given filtering method was calculated as the Bray-Curtis dissimilarity between the raw and the filtered datasets, after taxonomically classifying their reads down to the genus level.

In the even datasets, which contain the same number of 16S rRNA gene copies for each organism, all methods resulted in more than 20 OTUs after clustering. This was not surprising, since contaminations, PCR errors and sequencing errors were expected to inflate the observed diversity. In the staggered communities, in which the number of 16S rRNA gene copies varied by several orders of magnitude between the different organisms, the observed diversity was generally lower, due to some species being present at very low abundances. The total number of reported OTUs greatly varied between filtering methods, with Poisson binomial filtering consistently resulting in values that were the closest to the true diversity of the samples.

PBF also produced the highest proportion of accurate OTUs in all the 16S mock datasets for both sequencing platforms, while minimizing the number of singletons and spurious OTUs retrieved (Figure 2a,b). In the 454 and IonTorrent datasets, it also discarded the smallest number of reads and resulted in the smallest taxonomic bias (Figure 2c,d). In the Illumina datasets QIIME retrieved a larger number of reads, while both QIIME and *mothur* caused smaller taxonomic biases than our method. (Figure 2c,d - Illumina). In the case of QIIME, we believe that this was the result of a too shallow filtering, since it produced a remarkably lower proportion of accurate OTUs and a larger number of OTUs and singletons (Figure 2a,b - Illumina). As for *mothur*, their method filters sequences based on the presence of mismatches of similar quality scores in aligned paired reads (20), which likely makes it less susceptible to biases in quality distribution between different taxonomic groups (Supplementary Note 2).

The two filtering algorithms included in the USEARCH suite showed an intermediate performance in terms of the number and accuracy of the OTUs retrieved for both the 454 and Illumina platforms. Quality trimming yielded the smallest number of reads and resulted in the highest taxonomic bias, which supports the idea that over-stringent filtering may lead to undesirable effects. In the IonTorrent datasets, USEARCH filtering performed below Poisson binomial filtering for all the studied benchmarks (Figure 2 - IonTorrent). Finally, the *mothur* implementation of the PyroNoise algorithm (12) showed lower OTU accuracy than the other methods tested for filtering 454 reads. It has been previously described that the denoising process can introduce minor alterations in the original reads (28), a phenomenon that might explain these results. It must be noted that, albeit a pipeline for filtering IonTorrent reads with PyroNoise has been described, the IonTorrent mock community datasets were only available in Fastq format (S. Salipante, personal communication), which precluded the use of flowgram denoising algorithms. However, this limitation was not present for the environmental datasets, and a comparison of quality filtering algorithms for IonTorrent datasets that includes PyroNoise can therefore be found in Supplementary Figure SN5.3.

It should be noted that both the QIIME and USEARCH-UPARSE pipelines include specific post-clustering steps that would have improved the results obtained by their filtering methods alone. QIIME recommends to apply a post-hoc OTU size cut-off to reduce the retrieval of spurious diversity (21) at the cost of sensitivity. USEARCH-UPARSE,

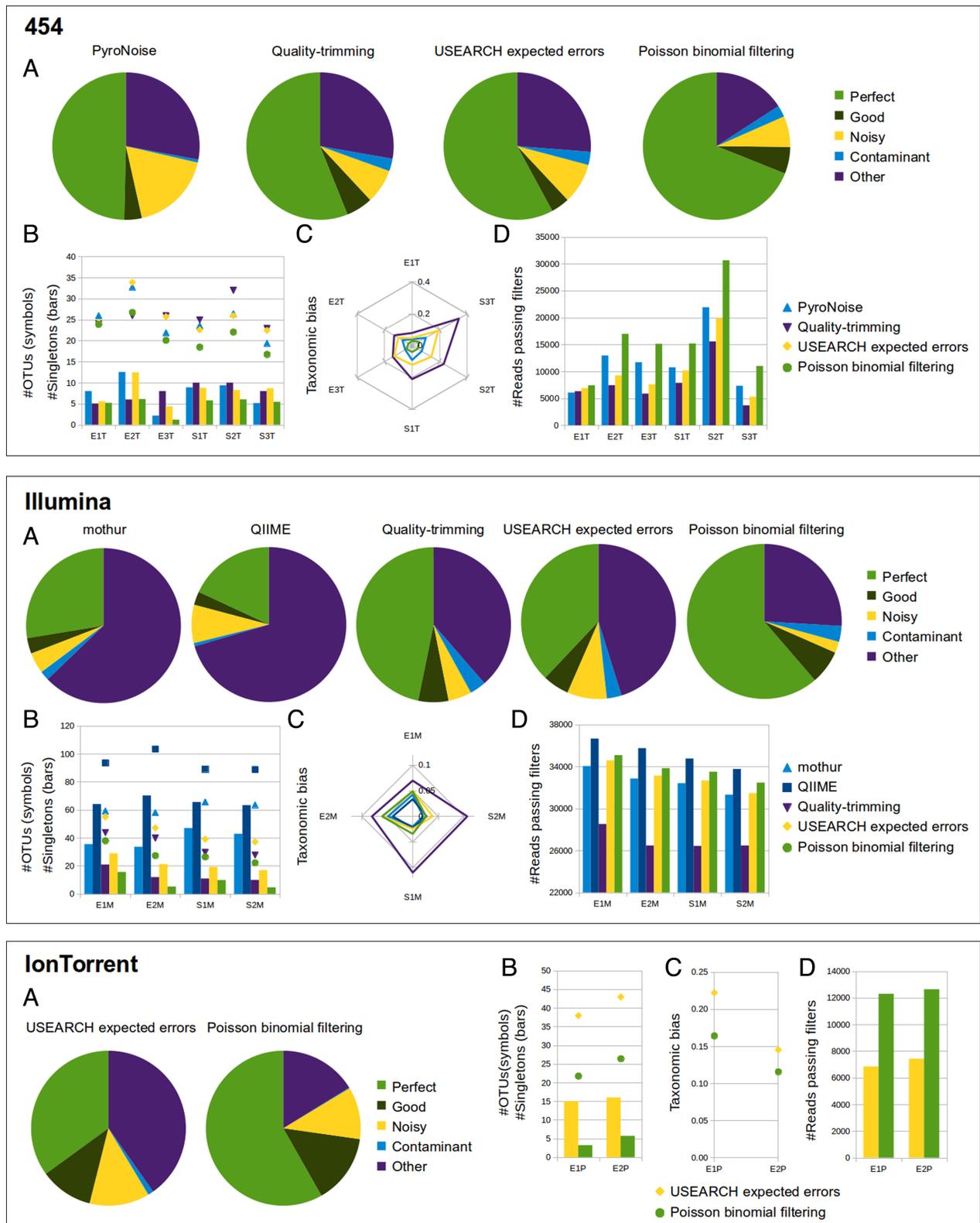


Figure 2. Comparison of filtering methods on 16S mock communities sequenced with the 454 GS FLX Titanium, Illumina MiSeq platforms and IonTorrent PGM platforms. (A) Pie charts constructed by averaging the fraction of OTUs on each accuracy category along the six 454 or the four Illumina samples. (B, D) Number of singletons (B, bars), total species (B, symbols) and reads (D) retrieved after filtering the raw reads with the different methods and performing chimera removal and clustering with a common pipeline. OTU and singleton numbers were obtained by averaging the results from 100 independent library size standardizations. (C) Taxonomic bias caused by the different filtering methods, measured as the Bray-Curtis dissimilarity between the raw and the filtered read communities.

in turn, maps the unfiltered reads to the OTU representative sequences after clustering in order to make an OTU table, and recovers reads with low predicted quality that are inferred to be good by comparison with high-quality sequences (15), increasing the number of retrieved reads. These steps are unrelated to quality-filtering *per se*, and can be applied independently of the method used to filter the sequences. Therefore, they were not included in our common post-filtering pipeline.

Performance of the different filtering methods on environmental datasets

The performance of the different filtering methods was also evaluated by quality-filtering publicly available datasets obtained by sequencing environmental communities (Supplementary Note 5). The results were similar to the ones obtained with the mock communities, with Poisson binomial filtering being the most consistent method in producing the smallest number of OTUs and singletons. Additionally, the OTUs obtained with PBF were overall the most similar to the 16S sequences present in the SILVA 16S reference alignment (25), which suggests that they contained the smallest number of errors. In the environmental 454 datasets, PyroNoise showed better results than in the 454 mock communities, but did it in an irregular fashion, especially in terms of OTU accuracy (Supplementary Figure SN5.1d). This inconsistency may be again due to the alteration of the original reads, and suggests that PyroNoise requires a finer parameter optimization than other approaches in order to be fully effective. In the environmental IonTorrent datasets PyroNoise discarded the smallest number of reads, but resulted in the highest number of singletons and OTUs, which also borne the least similarity to the reference alignment. USEARCH showed an intermediate performance between PyroNoise and Poisson binomial filtering (Supplementary Figure SN5.3). Finally, in the environmental Illumina datasets all filtering methods showed a similar behavior to that in the mock communities (Supplementary Figure SN5.2).

Quality-filtering is an additional source of taxonomic bias in microbial ecology studies

Even though the major sources of taxonomic biases in marker-gene-based studies are often related to differences in rRNA operon copy number, PCR and library construction (29–33), the read filtering process can increase this problem (Supplementary Note 2). We found significant biases in length and quality distribution between raw reads coming from different taxa in the mock 454 datasets (Figure 3a–c). Trimming them to a fixed length generated an artificial enrichment of the taxa with longer reads (Figure 3b), but since there is a decrease in quality at the end of 454 reads (see 15), it also resulted in a lower average read quality for the taxa with smaller raw reads (Figure 3d). This led to the generation of further taxonomic bias during the quality-filtering step (Figure 3f, Supplementary Note 2). Similar biases have been previously found in IonTorrent reads (34), and were confirmed during this study (Supplementary Note 2). Biases in read quality distribution between different taxa

were also found for the mock Illumina datasets, although to a lesser extent. Such biases were remarkably higher than those due to the random removal of reads (Supplementary Note SN2.1, Supplementary Note 6). We solved this problem by collapsing identical reads and choosing the one with the highest quality as a representative for filtering, in order to decide whether the whole group was discarded or allowed into the filtered dataset. This procedure reduced the effect of quality distribution biases, as even low abundance sequences are expected to have a high quality representative. Our solution rendered similar quality distributions for the different taxa, even after length trimming (Figure 3e,f), and significantly lower taxonomic biases than other filtering approaches, especially for 454 data (Figure 2c). Every method that relies on quality scores for sequence filtering will be affected by this source of bias. We therefore propose the approach described above as a general solution to this problem, since its simplicity makes it very easy to integrate into any filtering pipeline.

DISCUSSION

In this work, we have presented and validated the Poisson binomial algorithm for filtering sequence reads based on their error probability distributions. We have also demonstrated that Poisson binomial filtering is especially useful in the context of gene-marker-based studies, such as the study of microbial populations by amplifying and sequencing their 16S rRNA gene.

We compared our algorithm with other five quality-filtering methods that are included as defaults in mainstream pipelines such as mothur, QIIME or USEARCH, by analyzing mock and environmental datasets generated with three different sequencing platforms. Our results show that, when coupled to a standard analysis pipeline that included chimera removal and clustering, PBF proved to be the most accurate algorithm for filtering marker-gene sequences. While retaining a large number of sequences, it also resulted in OTUs that were the closest to the true biological species present in the studied samples, and minimized the generation of spurious diversity and taxonomic biases. These metrics are useful for algorithm benchmarking, especially when coupled to the analysis of mock communities, but they are not the typical end result of molecular ecology studies. Instead, users are typically interested in features such as relative community richness, or patterns of community composition. In Supplementary Note 7 we show that the choice of filtering method can also have substantial effects on the retrieved relative community richness, potentially affecting the final ecological interpretation obtained from a given dataset.

Remarkably, this algorithm does not rely on any particular error model. Instead, it just derives the error probability distribution of a given sequence from the quality scores of its individual bases. The only assumptions that our algorithm makes are that for any given sequencing platform, sequencing errors will be independent and the quality scores obtained during base calling will truly represent the probabilities of that base being wrong. This conceptual simplicity is one of its main advantages: as long as accurate quality scores are provided, Poisson binomial filtering will work in

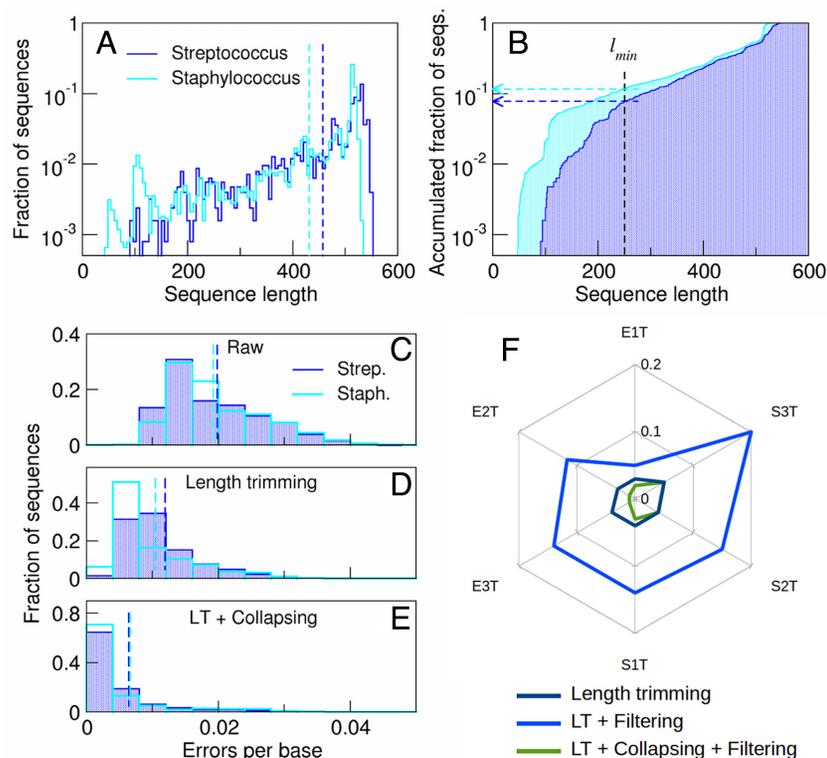


Figure 3. Addressing the taxonomic bias generated during the pre-processing and quality filtering of raw sequences. **(A, B)**: Raw reads from *Streptococcus* and *Staphylococcus*, the two most abundant genera in sample Even3T, show different length distributions. The dashed vertical lines in **(A)** indicate the average read lengths. The arrows in **(B)** indicate the fraction of reads from each taxon removed after discarding sequences shorter than $l_{\min} = 250$ nt. **(C, D, E)**: Errors per base distributions of *Streptococcus* and *Staphylococcus* reads in the **(C)** raw dataset, **(D)** after trimming the reads to 250 nt and discarding the ones shorter than the cut-off, and **(E)** after collapsing the trimmed reads. The dashed lines indicate average errors per base. Note that length trimming substantially increases the difference between the *Streptococcus* and *Staphylococcus* error distributions **(D)** when compared to that of the raw reads **(C)**. Filtering at this point would cause a 56.2% overrepresentation of *Streptococcus* versus *Staphylococcus* (see text and Supplementary Note 2). Collapsing identical reads prior to filtering solves this problem **(E)**, reducing the overrepresentation to 1%. **(F)**: Compositional bias generated during the pre-processing and filtering of the six 454 mock community samples, measured as the Bray-Curtis dissimilarity between the raw and the processed read communities. This shows that results in **(C, D, E)** can be generalized to all the taxa present in all the samples.

any present or future sequencing platform, with no need for further modifications.

In practice, quality-score calling ultimately depends on the sequencing platform manufacturer (454 quality-scores, e.g., do not predict per-base error probabilities, but the probability of overestimating the homopolymer length), and its accuracy is also influenced by the choice of primers and library preparation methods (35,36). Nonetheless, we have shown that, for the three sequencing platforms studied in this work, Poisson binomial filtering was able to correctly discriminate between good and erroneous sequences based solely on quality score information.

The fact that our method only relies on quality scores means that it will only account for sequencing errors, but not other errors such as PCR substitutions. However, it has been described that sequencing errors are responsible for the majority of singletons generated in molecular ecology studies (15,17). PCR chimeras are other source of spurious diversity, but dedicated algorithms such as UCHIME are able to accurately detect a large majority of them (see 15 for a more thorough discussion on this subject).

During the course of this research, we have also focused on a source of taxonomic bias that may have affected the

results of many molecular ecology studies. Notwithstanding the fact that upstream processes might introduce large biases in the amplicon pool to be sequenced, most of the methods used for filtering and analyzing marker-gene reads operate under the implicit (or even explicit, see 17) assumption that the probability of having j errors is the same for all reads, regardless of their origin. However, sequences from different taxa may have different length (for 454 and IonTorrent) and quality (for 454, IonTorrent and Illumina) distributions. This leads to the artificial enrichment of some taxa versus others during the quality filtering step, potentially compromising the quantitative interpretation of molecular ecology results obtained by high-throughput sequencing of marker-gene sequences. These biases are likely originated during base/quality calling: for instance, 454 reads show a systematic decrease in quality after homopolymer regions (37), which will penalize the taxa with longer homopolymer stretches on its 16S gene. We have nonetheless demonstrated that collapsing identical reads before the quality-filtering step greatly mitigates this issue.

In summary, the methodologies presented in this work substantially improve the existing filtering approaches in terms of OTU accuracy, observed alpha diversity and ob-

served community composition, delivering a more faithful representation of the original microbial communities present in the studied samples. Our algorithm is fast, easy to implement and works for every sequencing platform, constituting a valuable addition to all the existing pipelines for analyzing microbial ecology data.

SUPPLEMENTARY DATA

[Supplementary Data](#) are available at NAR Online.

ACKNOWLEDGEMENTS

The authors thank G. Ackermann, N.A. Bokulich, J.G. Caporaso and J.R. Rideout for facilitating access to the MiSeq mock community datasets, S. Salipante for providing information on the IonTorrent mock community datasets, P.S. Schloss for his insights into mothur's implementation of the Needleman-Wunsch algorithm, and A. Arce-Rodríguez, M.J. Gómez, J. Iranzo, S. Lincoln, F. López de Saro, S. Manrubia and C. Pedrós-Alió for useful feedback and discussions of the draft manuscript.

Author Contributions: F.P.S. and J.A. conceived the algorithm. J.A. developed the analytical work. F.P.S. and V.P. conceived the experiments. F.P.S. implemented the algorithm and designed and performed the experiments. F.P.S., J.A. and V.P. participated in the motivation and discussion of the results and contributed to the writing of the manuscript.

FUNDING

Subdirección General de Proyectos de Investigación of Spanish Ministerio de Economía y Competitividad [AYA2011-24803, FIS2011-27569, FIS2014-57686]; Spanish Comunidad de Madrid [MODELICO-CM-S2009ESP-1691]; European Research Council [250350]; computational resources were provided by the Data Intensive Academic Grid, which is supported by the United States National Science Foundation [0959894]; Spanish Consejo Superior de Investigaciones Científicas (CSIC) [to F.P.S. JAE-pre fellowship]. Funding for open access charge: Subdirección General de Proyectos de Investigación of Spanish Ministerio de Economía y Competitividad [AYA2011-24803].

Conflict of interest statement. None declared.

REFERENCES

- Roesch, L.F., Fulthorpe, R.R., Riva, A., Casella, G., Hadwin, A.K., Kent, A.D., Daroub, S.H., Camargo, F.A.O., Farmerie, W.G. and Triplett, E.W. (2007) Pyrosequencing enumerates and contrasts soil microbial diversity. *ISME J.*, **1**, 283–290.
- Costello, E., Lauber, C. L., Hamady, M., Fierer, N., Gordon, J.I. and Knight, R. (2009) Bacterial community variation in human body habitats across space and time. *Science*, **326**, 1694–1697.
- Caporaso, J.G., Lauber, C.L., Walters, W.A., Berg-Lyons, D., Lozupone, C.A., Turnbaugh, P.J., Fierer, N. and Knight, R. (2011) Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. *Proc. Natl. Acad. Sci. USA*, **108**, 4516–4522.
- Yatsusenko, T., Rey, F.E., Manary, M.J., Trehan, I., Dominguez-Bello, M.G., Contreras, M., Magris, M., Hidalgo, G., Baldassano, R.N., Anokhin, A.P. *et al.* (2012) Human gut microbiome viewed across age and geography. *Nature*, **486**, 222–227.
- Ding, T. and Schloss, P. (2014) Dynamics and associations of microbial community types across the human body. *Nature*, **509**, 357–360.
- Hugoni, M., Taib, N., Debroas, D., Domaizon, I., Jouan Dufournel, I., Bronner, G., Salter, I., Agogué, H., Mary, I. and Galand, P.E. (2013) Structure of the rare archaeal biosphere and seasonal dynamics of active ecotypes in surface coastal waters. *Proc. Natl. Acad. Sci. USA*, **110**, 6004–6009.
- Kunin, V., Engelbrektsen, A., Ochman, H. and Hugenholtz, P. (2010) Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. *Environ. Microbiol.*, **12**, 118–123.
- Schloss, P., Westcott, S.L., Ryabin, T., Hall, J.R., Hartmann, M., Hollister, E.B., Lesniewski, R.A., Oakley, B.B., Parks, D.H., Robinson, C.J. *et al.* (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.*, **75**, 7537–7521.
- Caporaso, J.G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F., Costello, E.K., Fierer, N., González-Peña, A., Goodrich, J.K., Gordon, J.I. *et al.* (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods*, **7**, 335–336.
- Edgar, R. (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, **26**, 2460–2461.
- Huang, Y., Niu, B.F., Gao, Y., Fu, L.M. and Li, W.Z. (2010) CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, **26**, 680–682.
- Quince, C., Lanzen, A., Davenport, R.J. and Turnbaugh, P.J. (2011) Removing noise from pyrosequenced amplicons. *BMC Bioinformatics*, **12**, 38.
- Schloss, P., Gevers, D. and Westcott, S. (2011) Reducing the effects of PCR amplification and sequencing artifacts on 16S rRNA-based studies. *PLOS One*, **6**, e27310.
- Caporaso, J.G., Lauber, C.L., Walters, W.A., Berg-Lyons, D., Huntley, J., Fierer, N., Owens, S.M., Betley, J., Fraser, L., Bauer, M. *et al.* (2012) Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. *ISME J.*, **6**, 1621–1624.
- Edgar, R. (2013) UPARSE: highly accurate OTU sequences from microbial amplicon reads. *Nat. Methods*, **10**, 996–998.
- Preheim, S.P., Perrotta, A.R., Martin-Platero, A.M., Gupta, A. and Alm, E.J. (2013) Distribution-based clustering: using ecology to refine the operational taxonomic unit. *Appl. Environ. Microbiol.*, **79**, 6593–6603.
- Tikhonov, M., Leach, R.W. and Wingreen, N.S. (2014) Interpreting 16S metagenomic data without clustering to achieve sub-OTU resolution. *ISME J.*, epub ahead of print 11 July 2014, doi:10.1038/ismej.2014.117.
- Quince, C., Lanzén, A., Curtis, T.P., Davenport, R.J., Hall, N., Head, I.M., Read, L.F. and Sloan, W.T. (2009) Accurate determination of microbial diversity from 454 pyrosequencing data. *Nat. Methods*, **6**, 639–641.
- Reeder, J. and Knight, R. (2010) Rapidly denoising pyrosequencing amplicon reads by exploiting rank-abundance distributions. *Nat. Methods*, **7**, 668–669.
- Kozich, J., Westcott, S., Baxter, N., Highlander, S. and Schloss, P. (2013) Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq Illumina sequencing platform. *Appl. Environ. Microbiol.*, **79**, 5112–5120.
- Bokulich, N., Subramanian, S., Faith, J.J., Gevers, D., Gordon, J.I., Knight, R., Mills, D.A. and Caporaso, J.G. (2013) Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. *Nat. Methods*, **10**, 57–59.
- Butler, K. and Stephens, M. (1993) The distribution of a sum of binomial random variables. Tech. Rep. Stanford University 467, Stanford, USA.
- Hildebrand, F., Tadeo, R., Voigt, A.Y., Bork, P. and Raes, J. (2014) LotuS: an efficient and user-friendly OTU processing pipeline. *Microbiome*, **2**, 30.
- Haas, B.J., Gevers, D., Earl, A.M., Feldgarden, M., Ward, D.V. and Giannoukos, G. (2011) Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Res.*, **21**, 494–504.
- Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J. and Glöckner, F.O. (2013) The SILVA ribosomal RNA gene

- database project: improved data processing and web-based tools. *Nucleic Acids Res.*, **41**, 590–596.
26. Edgar, R.C. and Flyvbjerg, H. (2015) Error filtering, pair assembly and error correction for next-generation sequencing reads. *Bioinformatics*, **31**, 3476–3482, btv401.
 27. Edgar, R., Haas, B., Clemente, J., Quince, C. and Knight, R. (2011) UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics*, **27**, 2194–2200.
 28. Gaspar, J. and Thomas, W. (2013) Assessing the consequences of denoising marker-based metagenomic data. *PLoS One*, **8**, e60458.
 29. Klappenbach, J.A., Dunbar, J.M. and Schmidt, T.M. (2000) rRNA operon copy number reflects ecological strategies of bacteria. *Appl. Environ. Microbiol.*, **66**, 1328–1333.
 30. Polz, M. and Cavanaugh, C. (1998) Bias in template-to-product ratios in multitemplate PCR. *Appl. Environ. Microbiol.*, **64**, 3724–3730.
 31. Klindworth, A., Pruesse, E., Schweer, T., Peplies, J., Quast, C., Horn, M. and Glöckner, F.O. (2013) Evaluation of general 16S ribosomal RNA gene PCR primers for classical and next-generation sequencing-based diversity studies. *Nucleic Acids Res.*, **41**, e1.
 32. Ross, M., Russ, C., Costello, M., Hollinger, A., Lennon, N.J., Hegarty, R., Nusbaum, C. and Jaffe, D.B. (2013) Characterizing and measuring bias in sequence data. *Genome Biol.*, **14**, R51.
 33. Turnbaugh, P.J., Quince, C., Faith, J.J., McHardy, A.C., Yatsunenko, T., Niazi, F., Affourtit, J., Egholm, M., Henrissat, B., Knight, R. *et al.* (2010) Organismal, genetic, and transcriptional variation in the deeply sequenced gut microbiomes of identical twins. *Proc. Natl. Acad. Sci. USA*, **107**, 7503–7508.
 34. Salipante, S.J., Kawashima, T., Rosenthal, C., Hoogstraal, D.R., Cummings, L.A., Sengupta, D.J., Harkins, T.T., Cookson, B.T. and Hoffman, N.G. (2014) Performance comparison of Illumina and Ion Torrent next-generation sequencing platforms for 16S rRNA-based bacterial community profiling. *Appl. Environ. Microbiol.*, **80**, 7583–7591.
 35. Loman, N.J., Misra, R.V., Dallman, T.J., Constantinidou, C., Gharbia, S.E., Wain, J. and Pallen, M.J. (2012) Performance comparison of benchtop high-throughput sequencing platforms. *Nat. Biotechnol.*, **30**, 434–439.
 36. Schirmer, M., Ijaz, U.Z., D'Amore, R., Hall, N., Sloan, W.T. and Quince, C. (2015) Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic acids res.*, **43**, e37.
 37. Brockman, W., Alvarez, P., Young, S., Garber, M., Giannoukos, G., Lee, W.L., Russ, C., Lander, E.S., Nusbaum, C. and Jaffe, D.B. (2008) Quality scores and SNP detection in sequencing-by-synthesis systems. *Genome Res.*, **18**, 763–770.