AEM
Journals.ASM.org

Distribution-Based Clustering: Using Ecology To Refine the Operational Taxonomic Unit

Sarah P. Preheim,[a] Allison R. Perrotta,[b] Antonio M. Martin-Platero,[b] Anika Gupta,[b] Eric J. Alm[a]

Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA[a]; Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA[b]

16S rRNA sequencing, commonly used to survey microbial communities, begins by grouping individual reads into operational taxonomic units (OTUs). There are two major challenges in calling OTUs: identifying bacterial population boundaries and differentiating true diversity from sequencing errors. Current approaches to identifying taxonomic groups or eliminating sequencing errors rely on sequence data alone, but both of these activities could be informed by the distribution of sequences across samples. Here, we show that using the distribution of sequences across samples can help identify population boundaries even in noisy sequence data. The logic underlying our approach is that bacteria in different populations will often be highly correlated in their abundance across different samples. Conversely, 16S rRNA sequences derived from the same population, whether slightly different copies in the same organism, variation of the 16S rRNA gene within a population, or sequences generated randomly in error, will have the same underlying distribution across sampled environments. We present a simple OTU-calling algorithm (distribution-based clustering) that uses both genetic distance and the distribution of sequences across samples and demonstrate that it is more accurate than other methods at grouping reads into OTUs in a mock community. Distribution-based clustering also performs well on environmental samples: it is sensitive enough to differentiate between OTUs that differ by a single base pair yet predicts fewer overall OTUs than most other methods. The program can decrease the total number of OTUs with redundant information and improve the power of many downstream analyses to describe biologically relevant trends.

Identifying meaningful operational taxonomic units (OTUs) is a significant bottleneck in the analysis of 16S rRNA sequences from complex microbial communities, particularly for large data sets generated by next-generation sequencing. Spurious sequences created by PCR or sequencing errors can greatly inflate the total number of OTUs (i.e., the alpha diversity) of a sample if not treated properly (1, 2). Although attempts have been made to address the problem of inflated alpha diversity from erroneous OTUs (1, 3–5), there have been few attempts to make OTUs that more accurately reflect ecologically cohesive bacterial populations.

Most common methods of forming OTUs with next-generation sequencing use a single genetic cutoff for creating OTUs. The most common approach for calling OTUs is to cluster sequences into groups based on sequence identity or genetic distances alone (taxonomy-independent [6], taxonomy-unsupervised [7], or *de novo* [8] clustering). Sequences are usually aligned using a pairwise or multiple-alignment algorithm to create a distance matrix, and sequences are clustered based on a sequence identity cutoff. Many heuristics have been developed to decrease the computational demand of OTU calling with various degrees of accuracy, such as CD-HIT (9), UCLUST (8), DySC (10), and ESPRIT (11). Another approach is to bin sequences into groups within a well-curated database of known sequences (taxonomy-dependent [6], phylotyping [12], or closed-reference [13] clustering). Sequences that do not match the database are lost, even though they could represent important, novel organisms. To overcome this problem, novel sequences can be retained as distinct clusters (open reference), but this comes at the expense of speed and convenience. All of these commonly applied techniques rely on a genetic cutoff, typically >97% sequence identity, to inform OTU clustering.

Although it is common to use a single sequence identity cutoff for clustering, more insight can be gained by adjusting the sequence clustering for individual taxonomic lineages (14, 15) or by using multiple genetic cutoffs for analysis (16, 17). Hunt et al. (14) developed a program called AdaptML to infer population boundaries from the ecological information on isolated strains. Different populations were often identified within what would generally be considered one species. Using two closely related populations predicted by AdaptML, Shapiro et al. (18) were able to investigate the early events of bacterial speciation. Koeppel et al. (15) used a program called EcoSim to infer units of bacterial diversity by estimating evolutionary parameters, such as periodic selection and drift, derived from phylogenetic relationships of isolated strains. This method can detect more total populations than are supported by AdaptML using ecology alone (19). Both Youngblut et al. (16) and Nemergut et al. (17) repeated their analyses at various levels of clustering. Youngblut et al. (16) found that using an inappropriate genetic cutoff would have changed their results. All of these studies demonstrate that more biological insight can be obtained from diversity studies when the clustering is done at different levels for different taxonomic lineages.

Sequencing and PCR errors and chimeras are significant issues in next-generation 16S rRNA libraries of microbial diversity. Inflated diversity estimates have been problematic with 454 pyrose-

Received 31 January 2013 **Accepted** 9 August 2013

Published ahead of print 23 August 2013

Address correspondence to Eric J. Alm, ejalm@mit.edu, or Sarah P. Preheim, spacocha@mit.edu.

Supplemental material for this article may be found at http://dx.doi.org/10.1128/AEM.00342-13.

Copyright © 2013, American Society for Microbiology. All Rights Reserved.

doi:10.1128/AEM.00342-13

quencing (1, 3–5, 20) and Illumina data sets (21, 22). Many attempts have been made to reduce the impact of sequencing error on the estimate of total diversity from chimeric sequences and PCR and sequencing errors (3–5). With good-quality filtering and strict error-correcting software, many errors can be detected and removed from the data set, reducing the effective error rate. However, these methods do not help in identifying how these "cleaned" sequences should be grouped into OTUs for downstream analyses.

We hypothesized that identifying the appropriate grouping for each taxonomic lineage and detecting many methodological errors can be accomplished using the distribution of sequences across samples. Bacteria in different populations will respond uniquely to variation in environmental conditions, resulting in different distributions across sampled environments. This has been demonstrated for different taxa under a range of conditions (14, 15) and during disturbance (16). Conversely, 16S rRNA sequences derived from the same population will have the same distribution across sampled environments, whether the sequences are from slightly different copies of the 16S rRNA gene in the same organism or variation of the 16S rRNA sequence within a population or are sequences generated randomly in error. Thus, whether the underlying distribution is the same for ecological (i.e., the same population of bacteria) or methodological (i.e., sequencing-error) reasons, they should be considered a group and merged into one OTU.

Our goal was to develop a simple algorithm using the distribution of 16S rRNA sequences across samples to inform the creation of OTUs for large next-generation sequencing studies. This method accommodates differences in the level of genetic differentiation across taxa and reduces the number of redundant OTUs from sequences within the same population or created by sequencing error. To apply this method to 16S rRNA surveys created from next-generation sequencing, we developed an algorithm that uses distribution information, the relative abundances of sequences within all samples, and genetic distance to inform clustering. We compare this method (distribution-based clustering [DBC]) to commonly applied closed-reference (i.e., phylotyping), open-reference (i.e., a hybrid of phylotyping and *de novo* clustering), and *de novo* clustering methods using experimental mock-community data sets. We test the accuracy and sensitivity of all clustering methods in identifying true input sequences, clustering sequencing and methodological errors with the input sequences they are derived from, and retaining the information contained in the distribution of sequences across samples. Distribution-based clustering reflects the true distribution of input templates or organisms more accurately than OTUs from methods using sequence identity alone. Finally, we compare the results of each clustering method on a set of unknown samples from a stratified lake, showing that DBC calls fewer OTUs than either the *de novo* or open-reference method yet is able to discriminate OTUs differing by a single base pair that show evidence of differing ecological roles. The source code, test data, and user guide are freely available for download at https://github.com/spacocha/Distribution-based-clustering.

MATERIALS AND METHODS

Previously generated mock community. We used an experimental mock data set that was previously generated (23) to test our clustering method. Data were downloaded from the supplemental data page of the Gordon laboratory website for the paper (http://gordonlab.wustl.edu/TurnbaughSE_2_10/PNAS_2010.html). The quality-filtered, denoised, and chimera-free data set was used for further analysis (http://gordonlab.wustl.edu/TurnbaughSE_2_10/Mock_nochimeras.fna.gz); all sequences were trimmed to 210 bases, and the first 14 bases were removed. The input sequences (http://www.w3.org/1999/xlink" xlink:href="http://gordonlab.wustl.edu/TurnbaughSE_2_10/MockIsolatesV2.fna.gz) and the input distributions from Table S3 in the supplemental material for reference 23 were also used in the analysis. Distribution information across samples was not included in the Mock_nochimeras.fna file, so it was derived from matching sequences in the cleaned data set (http://www.w3.org/1999/xlink" xlink:href="http://gordonlab.wustl.edu/TurnbaughSE_2_10/Mock_clean.fna.gz).

The representative sequence for *Providencia alcalifaciens* was mislabeled as *Providencia rettgeri*, as was evident from the distribution of the sequence across samples (which corresponded to the *P. alcalifaciens* distribution [see Fig. S1a in the supplemental material]), and matched many *P. alcalifaciens* strains in the NCBI nr database. The *P. rettgeri* sequence was replaced with the sequence from the data set that had the correct corresponding distribution (see Fig. S1b in the supplemental material) and that matched many *P. rettgeri* sequences in the NCBI nr database.

Mock-community generation. The second mock community used for much of this analysis was created from an environmental-clone library of 16S rRNA sequences from a lake sample. The DNA templates were 16S rRNA sequences on purified, linearized plasmids (i.e., Sanger clones) as described in the supplemental material, and approximately 800 bp was sequenced from the forward primer 27F (24). The input concentration of each DNA template was measured using a 2100 Bioanalyzer (Agilent Technologies Inc., Santa Clara, CA). DNA templates were mixed together into nine different mock communities ranging from simple (com1), with five DNA templates added, to complex (com9), with 40 total DNA templates. The DNA templates were mixed to create a range of final concentrations. Specific information about mock-community composition can be found in Tables S1 and S2 in the supplemental material.

Library construction and sequencing. Mock-community libraries for paired-end Illumina sequencing were constructed using a two-step 16S rRNA PCR amplicon approach diagramed in Fig. S2 in the supplemental material. The first-step primers (PE16S_V4_U515_F, 5′ ACACG ACGCT CTTCC GATCT YRYRG TGCCA GCMGC CGCGG TAA-3′; PE16S_V4_E786_R, 5′-CGGCA TTCCT GCTGA ACCGC TCTTC CGATC TGGAC TACHV GGGTW TCTAA T 3′) contain primers U515F and E786R targeting the V4 region of the 16S rRNA gene, as described previously (25). Additionally, a complexity region in the forward primer (5′-YRYR-3′) was added to help the image-processing software used to detect distinct clusters during Illumina next-generation sequencing. A second-step priming site is also present in both the forward (5′-ACACG ACGCT CTTCC GATCT-3′) and reverse (5′-CGGCA TTCCT GCTGA ACCGC TCTTC CGATC T-3′) first-step primers. The second-step primers incorporate the Illumina adapter sequences and a 9-bp barcode for library recognition (PE-III-PCR-F, 5′-AATGA TACGG CGACC ACCGA GATCT ACACT CTTTC CCTAC ACGAC GCTCT TCCGA TCT 3′; PE-III-PCR-001-096, 5′-CAAGC AGAAG ACGGC ATACG AGAT**N NNNNN NNN**CG GTCTC GGCAT TCCTG CTGAA CCGCT CTTCC GATCT 3′, where N indicates the presence of a unique barcode listed in Table S3 in the supplemental material).

Real-time PCR before the first-step PCR was done to ensure uniform amplification and avoid overcycling all templates. Both real-time and first-step PCRs were done similarly to the manufacture's protocol for Phusion polymerase (New England BioLabs, Ipswich, MA), as described in the supplemental material. Samples were divided into four 25-μl replicate reactions during both first- and second-step cycling reactions and cleaned using Agencourt AMPure XP-PCR purification (Beckman Coulter, Brea, CA). Environmental libraries were created as previously described using the two-step primer-skipping library protocol (26). The libraries were multiplexed together with other libraries not used in this study and sequenced using the paired-end approach on either the Genome Analyzer IIx or HiSeq 2000 Illumina sequencing machine at the BioMicro Center (Massachusetts Institute of Technology [MIT], Cambridge, MA). For environmental libraries and mock-community samples, respectively, 144 and 100 bases were sequenced from the forward and reverse orientations of the construct.

Pre- and postclustering quality control. Raw data were quality filtered using QIIME (version 1.3.0) (27) before processing with any clustering algorithm. The fastq files were processed using the split_library_fastq.py program of QIIME, truncating sequences when the base quality dropped below a Phred quality score of 17, which corresponds to a probability of error around 0.02 (using the command line options –last_bad_character Q -r 0). This quality filter stringency was chosen because it was found to result in the smallest Jensen-Shannon divergence (JSD) from the true distribution using com9 (see Fig. S3 in the supplemental material). Only sequences at least 99 bp long after quality filtering were retained (command line option -min_per_read_length 99). All other parameters were default parameters. After quality filtering, the complexity region between the adapters and the primer (see Fig. S2 in the supplemental material), along with the primer sequence, was removed using the trim.seqs program in mothur (version v.1.23.1) (28) and trimmed to 76 bp with a custom perl script (https://github.com/spacocha/Distribution-based-clustering/blob/master/bin/truncate_fasta.pl). All sequences not matching the first 15 bases of the primer were removed.

After each clustering algorithm, representative sequences were picked using QIIME pick_rep_set.py or a custom perl script (https://github.com/spacocha/Distribution-based-clustering/blob/master/bin/pick_most_ab_from_ablist.pl), using the most abundant sequence in the OTU as the representative. The sequences were used to determine which OTUs were correct (i.e., matched an input sequence) or incorrect (i.e., did not match an input sequence). OTUs were removed if the representative sequence did not align with the part of the 16S rRNA gene that was amplified (positions 13862 to 15958 of the Silva-based bacterial reference alignment; http://www.mothur.org/w/images/9/98/Silva.bacteria.zip) with at least 76 bp. OTUs with less than 2 counts or 11 counts were filtered out using QIIME's filter_otu_table.py (command line option -c 2 or -c 11) (see Table 2).

Closed-reference, open-reference, and *de novo* clustering methods. QIIME was used to make closed-reference (i.e., phylotype) and open-reference (i.e., a hybrid of phylotyping and *de novo* approaches) OTUs as described above. Closed- and open-reference clustering were done with the pick_reference_otus_through_otu_table.py flow from QIIME. Both methods used the 12_10 greengenes 97% reference OTU collection (ftp://greengenes.microbio.me/greengenes_release/gg_12_10/gg_12_10_otus.tar.gz) as the reference and UCLUST as the clustering algorithm (pick_otus:otu_picking_method uclust_ref), and new clusters were suppressed for closed-reference (pick_otus:suppress_new_clusters) but not for open-reference clustering. Example scripts are presented in the supplemental material.

mothur (v.1.23.1) (28) was used to form *de novo* OTUs using average neighbor hierarchical clustering following some of the standard protocol for processing 16S rRNA data (http://www.mothur.org/wiki/454_SOP). Sequences were aligned to the Silva reference alignment and trimmed using the align.seqs and screen.seqs/filter.seqs commands, respectively. A distance matrix was created and used to cluster the sequences for the calling of final OTUs using dist.seqs and cluster commands, respectively. A list of commands can be found in the supplemental material. The total numbers of OTUs were similar after chimera checking and lineage removal.

USEARCH (v. 6.0.307; drive5) was used to create the USEARCH *de novo* OTU with custom perl scripts for pre- and postprocessing, as described in the supplemental material, which are available at https://github.com/spacocha/Distribution-based-clustering/blob/master/bin.

Distribution-based clustering theory. Distribution-based clustering works by identifying bacterial populations at different levels of genetic differentiation for different taxonomic lineages by relying on the distribution of sequences across samples (i.e., the ecology) to determine where to draw population boundaries. Sequences that differ by only 1 base but that are found in different samples, suggesting they did not arise from the same underlying distribution, should be considered separately in downstream analyses and put into different OTUs (Fig. 1a). Conversely, 16S rRNA sequences drawn from the same underlying distribution across samples could be generated from differences between 16S rRNA operons in the same organism or variation of the 16S rRNA gene within a population or generated from random sequencing errors from a true sequence in the sample. These sequences should be grouped together and considered a unit (Fig. 1b). A statistical test (i.e., the chi-squared test) can be used to determine whether two sequences have similar distributions across libraries. Applying these metrics can merge sequences derived from the same population (e.g., sequencing error or interoperon variation) but retain ecologically distinct sequence types, even if they occur at the same genetic distance. It is important to note that the distribution-based approach will generate more spurious OTUs when sequencing errors are created in a nonrandom way across samples (i.e., higher error rates in a subset of libraries).

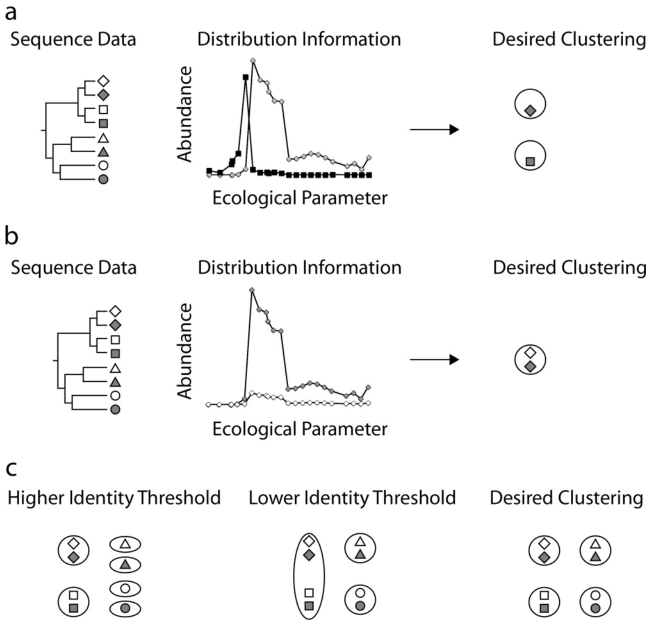

FIG 1 Schematic showing how the distribution-based clustering algorithm forms OTUs. Similar symbols represent sequences originating from the same template, organism, or population. Gray shading represents dominant sequences, and white represents low-abundance variants or errors. OTUs are represented as ovals or circles encompassing one or more symbols. (a) Hypothetical phylogenetic tree of the genetic relationship between various sequences represented by different symbols and shading. The distribution of two dominant sequences across one environmental parameter is shown. Using both the genetic and distribution information, distribution-based clustering identifies these as sequences originating from different organisms or populations and puts them in different OTUs. (b) Phylogenetic relationship and distribution of a dominant sequence and a low-abundance variant across some ecological parameter. Based on the sequence identity and distribution, distribution-based clustering merges these sequences in the same OTU. (c) Using genetic information alone, there is no way to achieve the desired clustering of sequences by symbol. Using a higher sequence identity cutoff will keep all dominant sequences in separate OTUs but will keep some low-abundance or erroneous sequences in different OTUs. Alternatively, using a lower identity cutoff, all low-abundance variants will be merged with the abundant variants, but the diamonds and squares are merged, as well.

Distribution-based clustering algorithm. Distribution-based clustering requires two input files, an OTU-by-library matrix and a distance

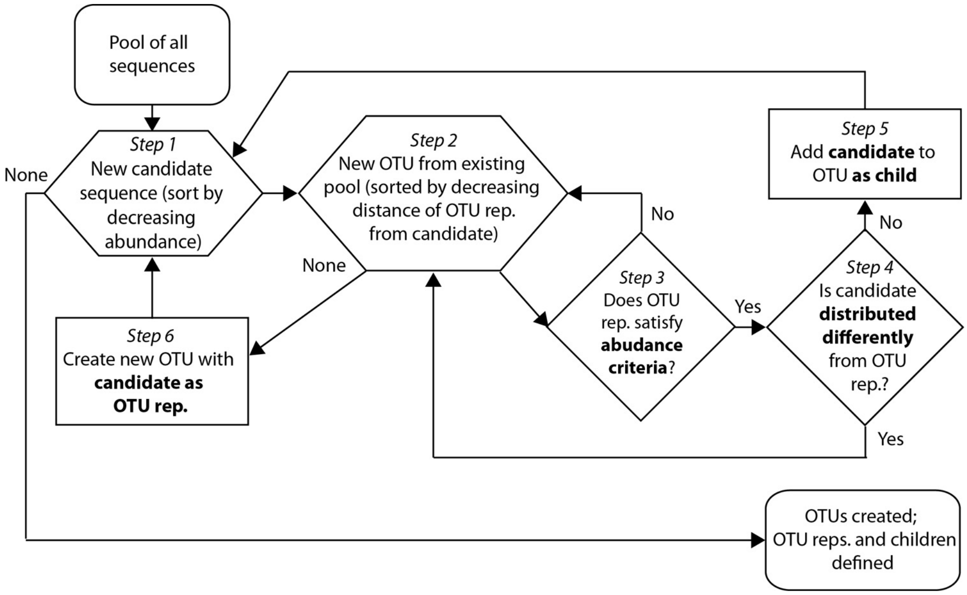
Pool of all sequences
Step 1
New candidate sequence (sort by decreasing abundance)
None
Step 2
New OTU from existing pool (sorted by decreasing distance of OTU rep. from candidate)
None
No
Step 3
Does OTU rep. satisfy abudance criteria?
Yes
Step 4
Is candidate distributed differently from OTU rep.?
No
Yes
Step 5
Add candidate to OTU as child
Step 6
Create new OTU with candidate as OTU rep.
OTUs created; OTU reps. and children defined

FIG 2 Outline of the decision-making process used during distribution-based clustering. The rounded rectangles indicate the beginning and end of the process, and the arrows point to the next step in the process. The hexagons indicate a loop, with the sorting criteria shown within the hexagon. The diamonds indicate a decision step, with the question contained within the diamond and arrows directing how the program will respond. The rectangles indicate action steps, where sequences are categorized as either representatives (rep.) of a new OTU or merged into an existing OTU.

matrix. Both the distribution and abundance are obtained from the OTU-by-library matrix. The distance matrix is important for ordering sequences according to increasing distance from the candidate sequences, as described below. Any method can be used to create a distance matrix. We use FastTree (29) with the -makematrix option, using both the aligned and unaligned sequences as inputs. This creates Jukes-Cantor-corrected distances and balances speed with accuracy. While this method works well on these mock communities, other distance matrices may be used as input, which may or may not improve accuracy.

OTUs are built in a stepwise manner (Fig. 2) in the following six steps. (i) Choose a candidate sequence. This sequence will either be added to an existing OTU or create a new OTU with itself as the representative, depending on the results of the subsequent steps. Consider candidate sequences from the pool of existing unique sequences in order of decreasing abundance. Abundance is defined as the number of times each sequence has been seen across all libraries. (ii) Choose an OTU from the pool of existing OTUs, sorted by decreasing distances of the representative sequence from the candidate. An OTU is evaluated if the representative sequence of the OTU is within the maximum genetic variation allowed to be within the same population (default -dist 0.1, the Jukes-Cantor-corrected distance of 0.1). Jukes-Cantor-corrected genetic distances were calculated using the -makematrix flag of FastTree (29), but other distance matrices can be used. The important information is the relative relationship of OTU representatives to the candidate sequence. Additionally, genetic distance is determined from the minimum of aligned and unaligned distances to reduce the impact of misalignment. If an OTU is found whose representative sequence is within the genetic-distance cutoff, proceed to step 3. Otherwise, stop the search and go to step 6. (iii) Determine whether the representative sequence of the candidate OTU satisfies the abundance criteria. The abundance of the representative sequence must be greater than a user-defined abundance threshold, defined as a *k*-fold increase over the abundance of the candidate sequence. To remove sequencing errors, thus creating OTUs that represent true sequences (not populations), a 10-fold abundance threshold is appropriate (-abund 10, default). This high abundance threshold restricts the total number of comparisons to OTUs with representatives that are much more abundant than the candidate sequences, which is common for sequences generated in error. To create OTUs that represent populations, a lower abundance threshold should be used, allowing comparisons with candidate sequences that are at an abundance similar to that of the OTU representative (-abund 0). This low abundance threshold provides the possibility to merge sequences together that were generated from interoperon variation or sequence variation with the population. If the representative sequence satisfies the abundance criteria, proceed to step 4. Otherwise, return to step 2 and choose another candidate OTU. (iv) Determine whether the candidate and representative sequences are distributed across samples in similar manners. The candidate sequence will be merged into the OTU unless there is evidence that its distribution is different from the distribution of the representative. The distributions of the candidate sequence (i.e., the observed distribution) and the OTU representative sequence (i.e., the expected distribution) are similar if the chi-squared test results in a *P* value above a user-defined cutoff (default = 0.0005). Sequences with low counts (e.g., singletons) will also result in high *P* values. *P* values are calculated using the R statistical language (chisq.test) or simulated (chisq.test:simulate.p.value) when the expected value is below 5 for more than 80% of the compared values. As an additional option, the JSD can be used. The JSD is commonly used to measure the distance between two distributions and can be applied when the difference between distributions is statistically significant but distributed in a similar manner (i.e., the chi-squared test is too sensitive). The JSD will commonly merge distributions that look similar by eye but are found to have statistically significant differences. However, it cannot be used as the sole metric, as it performs poorly on distributions with low counts. If the distributions are different, the next OTU is evaluated (step 2). Otherwise, proceed to step 5. (v) Add the sequence to the OTU. If the candidate sequence is distributed similarly to the representative sequence of the candidate OTU, the candidate sequence is added to the OTU and step 1 is repeated. (vi) Define OTU representatives. If none of the existing OTUs satisfy the criteria outlined above, an OTU is created with the candidate sequence as the representative of the OTU. This new OTU will not be merged with OTUs, but other sequences may be added.

Default parameters were chosen after varying each parameter in isolation and evaluated based on the total number of correct, merged, and incorrect-sequence OTUs (see Fig. S4 in the supplemental material). De-

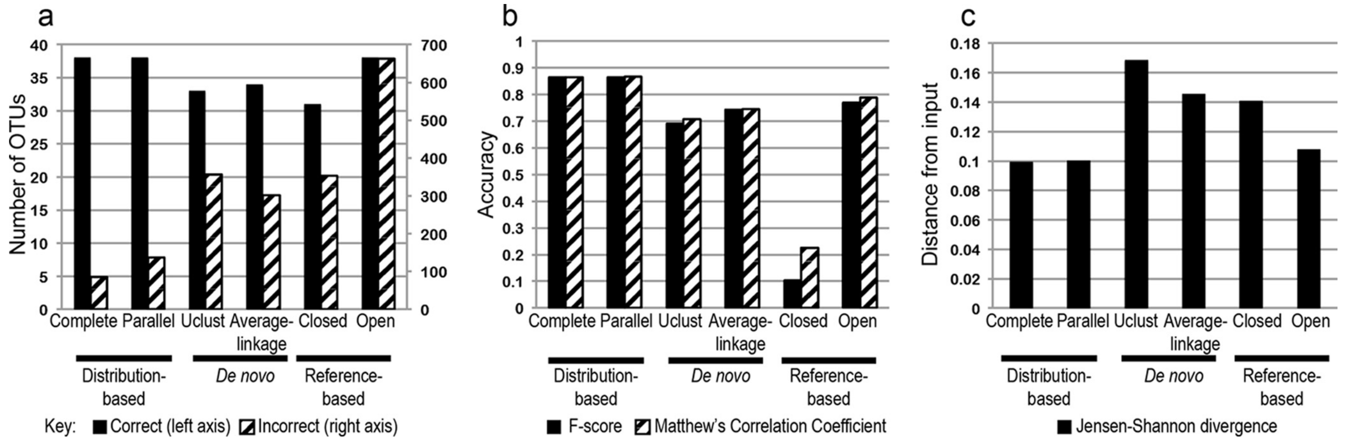

FIG 3 Distribution-based clustering results in more correct OTUs, fewer incorrect OTUs, and more accurately clustered reads originating from the same template in a mock community. (a) Total numbers of correct (left axis) and incorrect (right axis) OTUs predicted by each clustering method. A correct OTU is one in which the representative sequence matches one of the input sequences. (b) Accuracy of each clustering method at grouping together reads originating from the same template as measured by both the F-score and Matthew's coefficient correlation. (c) The JSD is used as a measure of distance from the input of resulting communities created by applying each clustering method.

fault parameters were used to cluster the mock-community sequences generated in this study. The previously generated, cleaned data set (23) was clustered with the following parameters: the distance cutoff was 0.05, the abundance criterion was 0, and the Jensen-Shannon divergence was used with a cutoff of 0.07. Ideally, these parameters would be optimized for different platforms.

Complete versus parallel algorithms. With the "complete" process, all sequences were analyzed together in the analysis. In the "parallel" process, sequences were preclustered with a heuristic approach (see below), and sequences in each cluster were processed separately, in parallel. However, sequences could be preclustered with different algorithms (e.g., nearest-neighbor single-linkage clustering), as long as the number of sequences that were grouped with their nearest neighbor was maximized. Data were preclustered with UCLUST into clusters for the new and previously generated mock communities, respectively, using a progressive clustering algorithm (https://github.com/spacocha/Distribution-based-clustering/blob/master/ProgressiveClustering.csh). Clustering was accomplished in several iterations by gradually relaxing the cutoff threshold. Sequences were first sorted by abundance and clustered with the UCLUST algorithm at 0.98 (1 bp difference is already below 0.99). The seeds of these clusters were sorted by abundance and clustered again at 0.97. This was repeated to the lowest threshold value of 0.9 for the mock communities generated in this study and 0.95 for the Turnbaugh et al. mock community (23). The resulting files were consolidated to make a list of clustered sequences. The distribution-based algorithm is used in parallel on sequences in these clusters. If the abundance of all members of the group is lower than the abundance threshold, the cluster remains intact (i.e., a low-count cluster with no information). However, the cluster is divided when two OTU representative sequences are identified.

Assessment of accuracy. We assessed how well the resulting OTUs represent the true input sequences. We expect sequences originating from the same input organism or template to be clustered together and sequences originating from different input organisms or templates to remain distinct, even with as little as 1 bp of difference between them. The corresponding input organisms or template for each resulting sequence was determined as the smallest distance (the minimum of aligned and unaligned distances) to an input sequence for each unique sequence. Sequences were weighted by abundance, so more abundant sequences resulted in more total counts.

To assess the accuracy of each method against our criteria, we used two measures of a test's accuracy, the F-score and the Matthew's correlation coefficient (MCC). True positives (TP) are defined as a pair of sequences in the same OTU originating from the same input organism or template. False positives (FP) are defined as a pair of sequences in different OTUs originating from the same input. True negatives (TN) are defined as a pair of sequences in different OTUs originating from different inputs. False negatives (FN) are defined as a pair of sequences in different OTUs originating from the same input or if either of a pair of reads was not assigned to an OTU (only affecting closed-reference clustering). These were calculated with various scripts using the resulting OTU list from each algorithm, along with a mapping file indicating the input (determined as described above) and a translation file mapping reads to libraries (https://github.com/spacocha/Distribution-based-clustering/tree/master/confusion_matrix_calc).

The F-score was calculated as follows: F-score $= 2 \times$ (precision $\times$ recall)/(precision + recall), where precision is defined as TP/(TP + FP) and recall is defined as TP/(TP + FN). The MCC was calculated as previously described (30): MCC $= (TP \times TN - FP \times FN)/\sqrt{[(TP + FN)(TP + FP)(TN + FP)(TN + FN)]}$, with TP, FP, TN, and FN as defined above.

Comparison with the input community. To compare the resulting OTU-by-library matrix with the expected distribution (see Table S3 of Turnbaugh et al. [23] and Table S2 in the supplemental material), we used the JSD from mock community com9 and Uneven2 library for the Turnbaugh et al. (23) data set for comparison. OTUs were paired to an input sequence through the sequence representative (i.e., the most abundant sequence in the OTU) with a match to an input sequence or by the most abundant OTU with a best Blast hit to the input organism. The total abundance of reads mapping to each OTU from com9 or Uneven2 was compared to the concentration of each corresponding mock-community member (Fig. 3c and 4c). The JSD was calculated with dist_mat (metric = JS) using PySurvey (https://bitbucket.org/yonatanf/pysurvey).

Nucleotide sequence accession numbers. All clone sequences were submitted to GenBank (accession no. KC192376 to KC192544). Illumina data were submitted to the Sequence Read Archive under study accession numbers SRP029590 (mock community) and SRP029470 (environmental sample).

RESULTS

Distribution-based clustering goals. Our goal was to develop a clustering algorithm that merges sequences derived from the same input organism or template but keeps separate those originating from different input organisms or templates (Fig. 1). Sequences derived from the same input could represent microdiversity from

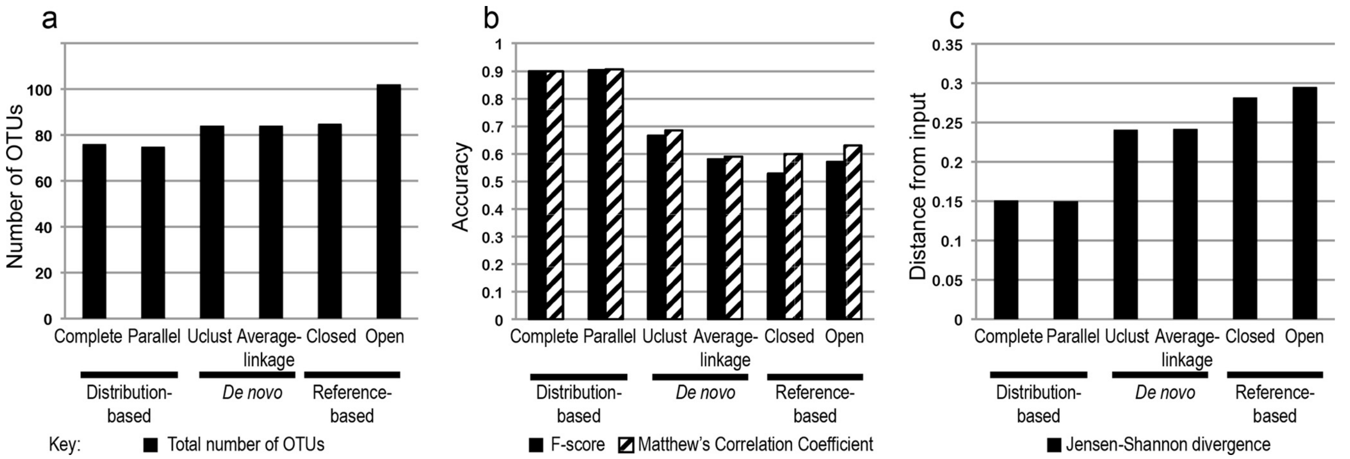

FIG 4 Distribution-based clustering predicts fewer OTUs and more accurately clustered reads originating from the same organism in a cleaned, denoised, and chimera-free mock community. (a) Total number of OTUs predicted by each clustering method. (b) Accuracy of each clustering method at grouping together sequences that originated from the same organism, as measured by both the F-score and Matthew's coefficient correlation. (c) The JSD is used as a measure of distance from the input of resulting communities created by applying each clustering method.

interoperon variation, closely related organisms within the community with highly similar functions and the same fitness across sampled environments, or sequencing error. However, we also wanted an algorithm that has the sensitivity to detect different populations, even if the similarity between sequences in different populations is greater than what is typically used for species designations (i.e., above 97% sequence identity) or within the range of sequencing error. We compare the resulting method using two different experimental mock communities to demonstrate how this algorithm compares to more commonly applied clustering methods based on sequence identity alone.

Distribution-based clustering more accurately clusters sequences created in error. Distribution-based clustering creates OTUs that more accurately represent the input sequences based on the total number of OTUs, how sequences are grouped together into OTUs, and the distribution of OTUs across samples. Thirty-eight mock template sequences remain in distinct OTUs in both distribution-based and open-reference clustering, resulting in the largest number of OTUs containing at least one of the input sequences (Fig. 3a, Correct). *De novo* clustering has fewer correct OTUs because some sequences are merged into the same OTU. Closed-reference clustering retains fewer correct OTUs because some of the community members do not match the database with sufficient identity. Distribution-based clustering predicts the lowest number of spurious, incorrect OTUs (Fig. 3a, Incorrect). Open-reference clustering predicts the largest number of incorrect OTUs of all methods.

Distribution-based clustering also groups reads originating from the same template sequence together more accurately. A typical benchmark of OTU accuracy is whether the algorithms cluster sequences that are within a specific genetic distance or sequence identity threshold (12). However, our benchmark is whether reads that originate from the same mock template are grouped together and reads originating from different templates are kept apart. The F-score and Matthew's correlation coefficient are both measures of classification accuracy that have been used previously to benchmark OTU definitions (12). By either metric, distribution-based clustering outperforms all of the other methods at accurately discriminating input sequences (Fig. 3b). *De novo* clustering predicts more true positives than distribution-based clustering but also predicts about 10 times more false positives than distribution-based clustering (Table 1) because it tends to overcluster the closely related true sequences. Closed-reference clustering has the lowest scores due to a large number of false negatives for sequences that do not match the database.

Distribution-based clustering produces a resulting community that is more similar to the input community in both the total number and relative abundance of OTUs. The number of reads mapping to each OTU from one high-quality library (com9) was compared to the input sequences using the Jensen-Shannon divergence (Fig. 3c). Distribution-based clustering (both complete and parallel applications [see "Complete versus parallel algorithms" above for details]) had the smallest Jensen-Shannon divergence from the input community of all clustering algorithms. Both *de novo* algorithms result in the largest divergence from the true distribution of all clustering methods because some input sequences are merged together. Closed-reference clustering discarded many input sequences that did not match the database, resulting in a larger calculated divergence from the input commu-

TABLE 1 Abilities of clustering algorithms to group reads from the same input sequence together into the same OTU

	No. predicted					
	Distribution based		*De novo*		Reference based	
Result[a]	Complete	Parallel	USEARCH	Avg[b]	Open	Closed
TP	8.57E8	8.45E8	9.46E8	8.60E8	6.61E8	6.68E8
FP	6.86E7	5.08E7	7.36E8	3.92E8	1.32E4	1.10E4
TN	1.48E11	1.48E11	1.48E11	1.48E11	1.48E11	1.37E11
FN	2.02E8	2.14E8	1.13E8	1.99E8	3.98E8	1.15E10

[a] TP, a pair of sequences in the same OTU with the same sequence of origin; FP, a pair of sequences in different OTUs with the same sequence of origin; TN, a pair of sequences in different OTUs with different sequences of origin; FN, a pair of sequences in different OTUs with the same sequence of origin or if either of a pair of reads was not assigned to an OTU (only affects closed-reference clustering).

[b] Avg, average-linkage hierarchical clustering.

TABLE 2 Total numbers of OTUs remaining after filtering out low-abundance OTUs

	No. of OTUs remaining[a]					
	Mock community			Environmental sample		
Method	No filter	>1	>10	No filter	>1	>10
DBC (complete)	124	82	63	NA	NA	NA
DBC (parallel)	175	136	83	14,234	11,762	6,087
De novo (USEARCH)	390	226	86	23,616	17,261	7,875
De novo (avg linkage)	336	169	70	NA	NA	NA
Closed reference	700	430	160	9,799	7,867	4,046
Open reference	385	257	119	23,047	15,833	6,310

[a] Filtering criteria: either all OTUs were included (No filter) or only OTUs with greater than 1 (>1) or greater than 10 (>10) counts were included. NA, not applicable.

nity. Open-reference clustering does not merge as many input sequences as *de novo* clustering and does not discard any true sequences like closed-reference clustering but was still less accurate than distribution-based clustering.

Filtering out low-abundance OTUs. Low-abundance OTUs are often discarded because they do not contain much information. We also compared the total numbers of OTUs remaining after filtering to various levels (Table 2). After filtering out singletons (i.e., OTUs with less than 2 counts), distribution-based clustering still predicts many fewer OTUs than any other method for the mock community and fewer than *de novo* and open-reference clustering in the environmental sample. However, the total numbers of OTUs are similar after filtering out OTUs with 10 or fewer counts.

DBC more accurately groups sequences from the same organism. The mock community generated by Turnbaugh et al. (23) provides the opportunity to highlight the power of this approach at grouping together sequences originating from the same organism while still keeping the power to resolve closely related organisms that have a unique distribution across samples. The input of this mock community came from DNA extracted from 67 organisms. The data in this analysis were previously cleaned and denoised, and chimeras were removed (23). Thus, the following results describe how well this method does at clustering sequences in the absence of sequence error.

Distribution-based clustering is better than other methods at merging together sequences that originated from the same input organism and accurately representing the input distribution. The complete and parallel versions of distribution-based clustering predicted 76 and 75 total OTUs, respectively, the smallest total number of OTUs of all clustering methods (Fig. 4a). It also more accurately grouped together reads that originated from the same organism (Fig. 4b) and more accurately captured the distribution of the input sequences (Fig. 4c). Closed- and open-reference clustering never grouped together sequences that originated from different organisms (i.e., no false positives) but did not merge as many sequences that originated from the same organism in the same OTUs (i.e., fewer true positives), not clustering together enough sequences (i.e., underclustering). Both *de novo* approaches tended to merge sequences originating from closely related organisms (i.e., more false positives) but also more often grouped together sequences from the same organism (i.e., more true positives), grouping together too many sequences (i.e., overclustering). These results highlight the drawback of using genetic information alone, which will necessarily either overcluster or undercluster sequences, as depicted in the example in Fig. 1c. Using the distribution of sequences across samples is the only way to cluster more sequences by their inputs when the levels of genetic variation are different across taxonomic lineages.

Comparison with unknown samples. Along with comparisons between clustering methods in a simple, well-defined mock community, we also applied all clustering methods to an environmental-sample set. This sample set was generated from 25 samples from a depth profile of a stratified lake sample (Mystic Lake, Winchester, MA), where different depths corresponded to distinct biogeochemical conditions. We generated two data sets for this analysis. First, we made an Illumina 16S rRNA library from DNA extracted from water collected approximately every meter from the surface to the bottom (22-m depth). Additionally, we generated Sanger-sequencing-based 16S rRNA clone libraries (Sanger data set) from two depths, 6 and 21 m (described in the supplemental material). The distribution of the Illumina library sequences was used in the clustering method, and Illumina sequences that matched different Sanger clones were used as a control comparison, since these sequences were observed independently in the Illumina and Sanger data sets.

Closed-reference clustering overfilters environmental data. The closed-reference clustering method predicts the smallest number of OTUs of all methods (Fig. 5a). Although the total number of OTUs in the sample is unknown, the Illumina sequences that match the Sanger library mock community can be used to compare clustering methods on the unknown sample; 89 Illumina sequences match one or more of the Sanger sequences. As we saw with the simple mock community, which was generated from clones of these sequences, the closed-reference method discards many sequences that are missing representative sequences in the database. Closed-reference clustering discards 15 of the 89 sequences with more than 1,000 counts across all libraries. The most abundant discarded sequence is classified as cyanobacteria, with a distribution that corresponds to a peak in oxygen below the thermocline. This suggests that the very low number of OTUs predicted by the closed-reference method is an underestimate and that the method excluded biologically interesting information.

Overclustered environmental data. *De novo* and open- and closed-reference clustering overclustered the data, resulting in skewed environmental distributions for many OTUs compared with distribution-based clustering. Merged sequences with different distributions produced low correlations between the resulting OTU and the matching Sanger clone for different clustering methods because merged sequences had very distinct profiles (e.g., Fig. 5b). The distribution of five OTUs formed by *de novo* (USEARCH) clustering resulted in correlations below 0.9 with the matching Sanger sequence (see Table S4 in the supplemental material). Three OTUs formed by open- and closed-reference clustering algorithms had low correlations with the matching Sanger sequence (see Table S4). However, the correlation of the matching Sanger sequence with distribution-based clustering OTUs was high in all cases. This suggests that other clustering methods are more likely to overcluster sequences with distinct environmental distributions than distribution-based clustering.

Distribution-based clustering is accurate and flexible. The

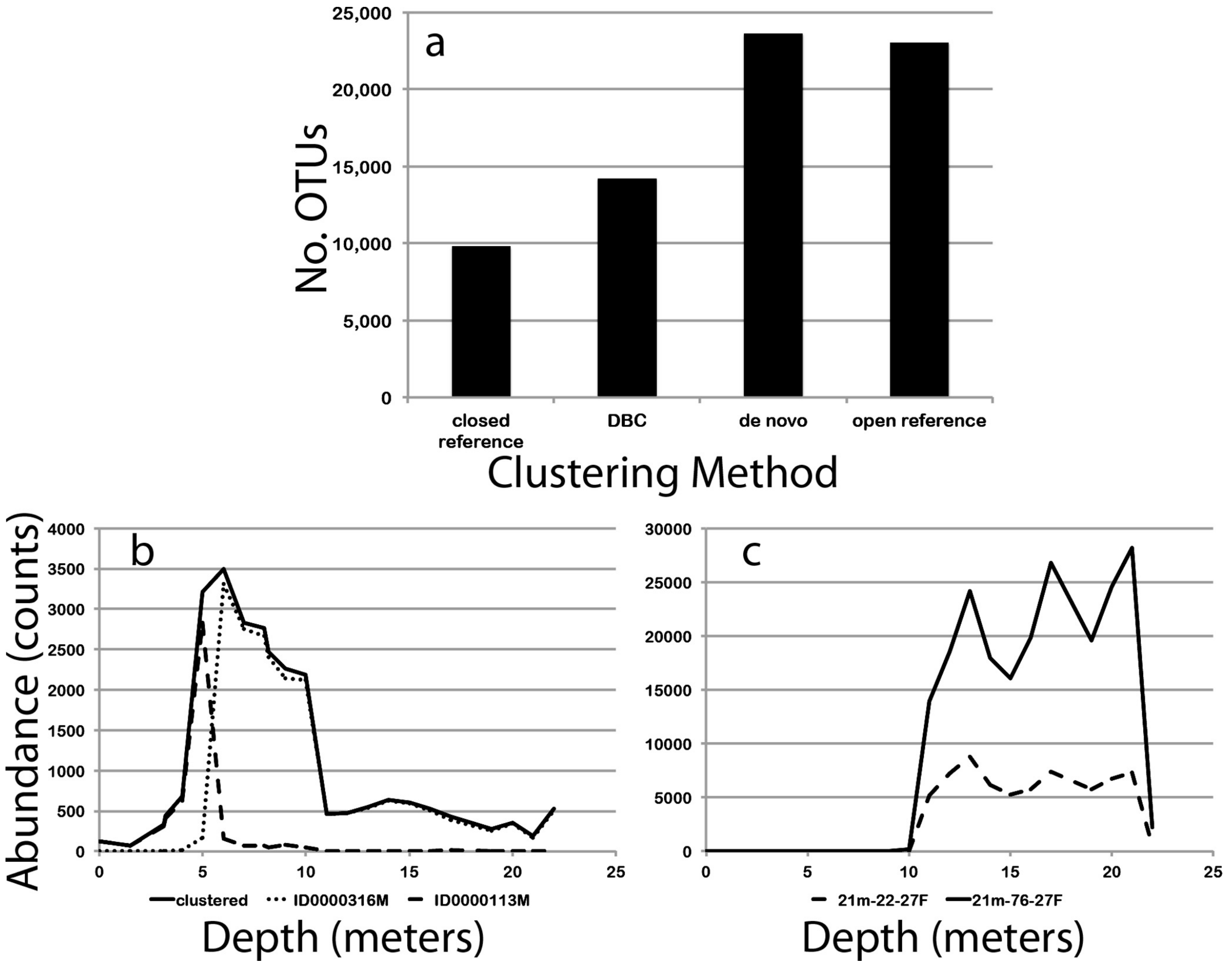

FIG 5 Evaluation of clustering methods on environmental samples from a stratified lake. (a) Total number of OTUs predicted by each clustering method for the entire lake. (b) Sequences displaying distinct ecological distributions but merged by all clustering algorithms except distribution-based clustering. The solid line is the distribution of the resulting cluster, comprised mainly of two sequences (dotted and dashed lines). Distribution-based clustering keeps the two sequences distinct, but all other methods merge them into one OTU. (c) Sequences that represent microdiversity within the environmental sample. The distribution-based clustering algorithm can be adjusted so that these sequences either remain distinct or can be clustered.

distribution-based clustering method predicted a low number of OTUs yet retained distinct profiles for highly similar sequences. Distribution-based clustering predicted about 9,000 fewer OTUs than both *de novo* and open-reference clustering (Fig. 5a). When singletons (i.e., OTUs with 1 count) were filtered out, distribution-based clustering still predicted several thousand fewer total OTUs than either *de novo* or open-reference clustering. However, after filtering out OTUs with less than 10 counts across all libraries, the difference was less obvious (Table 2). Distribution-based clustering was also sensitive enough to keep closely related sequences with distinct distributions in separate OTUs (Fig. 5b).

Distribution-based clustering can function to identify all likely sequences in the sample (i.e., remove sequencing error) or be used to group all sequences together that are within a population (i.e., ecologically relevant populations). To eliminate sequencing error, the representative sequence of the OTU must be at least 10-fold more abundant than other sequences in the OTU, since sequences created in error are typically less than 10% of the abundance of the original sequence. This is comparable to the analysis done with the mock community generated from 16S plasmid templates (Fig. 3). Under these conditions, the sequences shown in Fig. 5c would remain distinct OTUs. However, it may be redundant to consider each sequence a separate OTU because they are genetically similar and distributed in similar manners. Thus, the distribution-based algorithm can also be adjusted to merge the sequences in Fig. 5c by using no abundance cutoff and comparing the sequence distributions with the JSD (see Materials and Methods for details). This is comparable to the analysis done on the mock community generated from genomic DNA extracted from different organisms (Fig. 4). Under the adjusted parameters, distribution-based clustering predicted a total of 11,871 OTUs and created three OTUs with more than one sequence matching Sanger clones, including the sequences in Fig. 5c.

Run time of each clustering algorithm. The total computa-

TABLE 3 Representative clustering times for mock-community samples with various algorithms

	Total run time (h:min:s)[a]	
Clustering method	Mock community[b]	Environmental sample[c]
Distribution-based clustering (complete)	1:09:40	NA
Distribution-based clustering (parallel)[d]	0:21:31	7:58:57
De novo (avg neighbor)	0:06:36	NA
De novo (USEARCH)	0:00:23	0:00:26
Closed reference	0:06:09	1:26:23
Open reference	0:06:05	1:23:25

[a] Times are approximated by the difference between the start time and end time in the shell script examples in the supplemental material. NA indicates that the method was not performed.

[b] The mock community contains 565,498 total reads and 5,489 unique sequences.

[c] The environmental sample contains 7,539,779 total reads and 120,601 unique sequences.

[d] The distribution-based clustering algorithm was the only one that was parallelized; 60 to 100 different processes were run at one time. The other methods would have had improved speeds if run in parallel.

tional time for distribution-based clustering is much longer than that of any of the other clustering methods. Table 3 shows typical run times for approximately 500,000 total reads (5,489 unique sequences) in the mock community and 7.5 million reads (120,601 unique sequences) in the environmental sample. Only the parallelized distribution-based clustering used multiple processors to complete, and the run times of the other methods could be improved even further by using multiple processors. However, it is clear that there is a significant difference in speed between distribution-based clustering and the other methods.

Issues affecting sequence and distribution accuracy. The sequences and distribution of OTUs across libraries should represent the true distribution as accurately as possible. Recommendations made from previous studies were followed during library construction to reduce PCR amplification biases, including reducing the cycle number and pooling replicate PCRs (31, 32). While these measures help, the resulting sequences and distributions across libraries are primarily affected by two things: mismatches between the primer and template sequences and sequence-specific errors of the Illumina sequencing platform from a poor-quality run.

Sequence-specific sequencing errors. The distribution-based clustering method is sensitive to errors that are generated in a nonrandom way across samples. Since the algorithm assumes that differences in the distribution of sequences across samples represent important information, this assumption is invalid when differences are due to methodological errors. In our analysis, the most obvious cause of nonrandom errors is combining sequencing data from different runs with varying quality scores (see Fig. S5 in the supplemental material), as certain errors were generated at a higher frequency on one flow cell than the other (see Fig. S6a in the supplemental material). This causes the erroneous sequences to have a significantly different distribution than the sequences they were derived from (see Fig. S6b in the supplemental material), and they are thus retained as distinct OTUs. As expected, distribution-based clustering performs very well on simulated data when the error rate is constant across libraries but is substantially worse when error rates are nonconstant (see Table S5 in the supplemental material). Thus, distribution-based clustering would have been even more accurate had all of the samples been sequenced on the same flow cell.

Sequence-specific errors are obvious when a stringent quality filter is applied to a low-quality sequencing lane. After removing templates with primer site mismatches, Fig. 6 shows little decrease in the correlation between the observed and expected frequencies for a good-quality sequencing run after quality filtering (Fig. 6a and b). In a library from the poor-quality lane (flow 2, lane 1; com4 to com6), the correlation with the input concentration is high for unfiltered data ($R^2 = 0.96287$) (Fig. 6c). However, the correlation between the input concentration and the resulting sequences breaks down with more stringent quality filtering ($R^2 = 0.49601$) (Fig. 6d). This is likely due to sequence-specific errors, a problem identified previously with Illumina sequencing technology (33–35). When using data from poor-quality sequencing runs, OTUs from more stringent quality filtering represent true sequences, but the relative abundances may be highly skewed.

DISCUSSION

We present a novel method of calling OTUs that uses the ecology of the organisms they represent to inform the clustering. Typically, only genetic information is considered when forming OTUs. Incorporating information such as abundance and distribution into the OTU formation process creates OTUs that more accurately cluster sequences by the template or organism of origin and improves the information content of the resulting OTUs.

The gross trends in the data are similar, regardless of clustering algorithms. Principal-coordinate analysis (PCoA) plots, which identify the most obvious differences between samples, were similar across clustering methods (see Fig. S7 and S8 in the supplemental material). PCoA is particularly effective when the variable of interest (e.g., depth or disease state) is associated with major changes in community structure but is less effective at detecting subtle variations in community structure. Furthermore, it cannot pinpoint the specific sequences that drive these associations. Other approaches, such as univariate tests, including the Mann-Whitney U test and Fisher's exact test, and statistical learning techniques, such as random-forest classification, can test for associations between bacterial species abundance and environmental metadata (36). Optimizing the clustering algorithm to detect such associations will increase the chances of gaining important biological insight. Thus, accurate OTU formation may not be as critical when trends in the data can be discerned at higher taxonomic levels, such as the ratio of *Bacteroidetes* to *Firmicutes* in obesity (37). However, differences between closely related organisms are crucial for identifying evolutionary and ecological mechanisms (18). In such cases, distribution-based clustering may be one of only a few tools that can be used to distinguish the signal from the noise of sequencing errors.

Run time is currently a severe limitation to implementing distribution-based clustering on very large data sets. Although many improvements can be made to the algorithm itself to increase the speed of the program (likely with lower accuracy), any implementation will likely be more computationally intensive than other methods, since it involves processing additional information. Steps can be taken to reduce the total run time, such as increasing the abundance skew (e.g., 100-fold more abundant representative sequences), decreasing the total-distance cutoff allowed for forming clusters (e.g., a cutoff of 0.05), or filtering out low-abundance sequences (e.g., singletons). All of these steps decrease the total

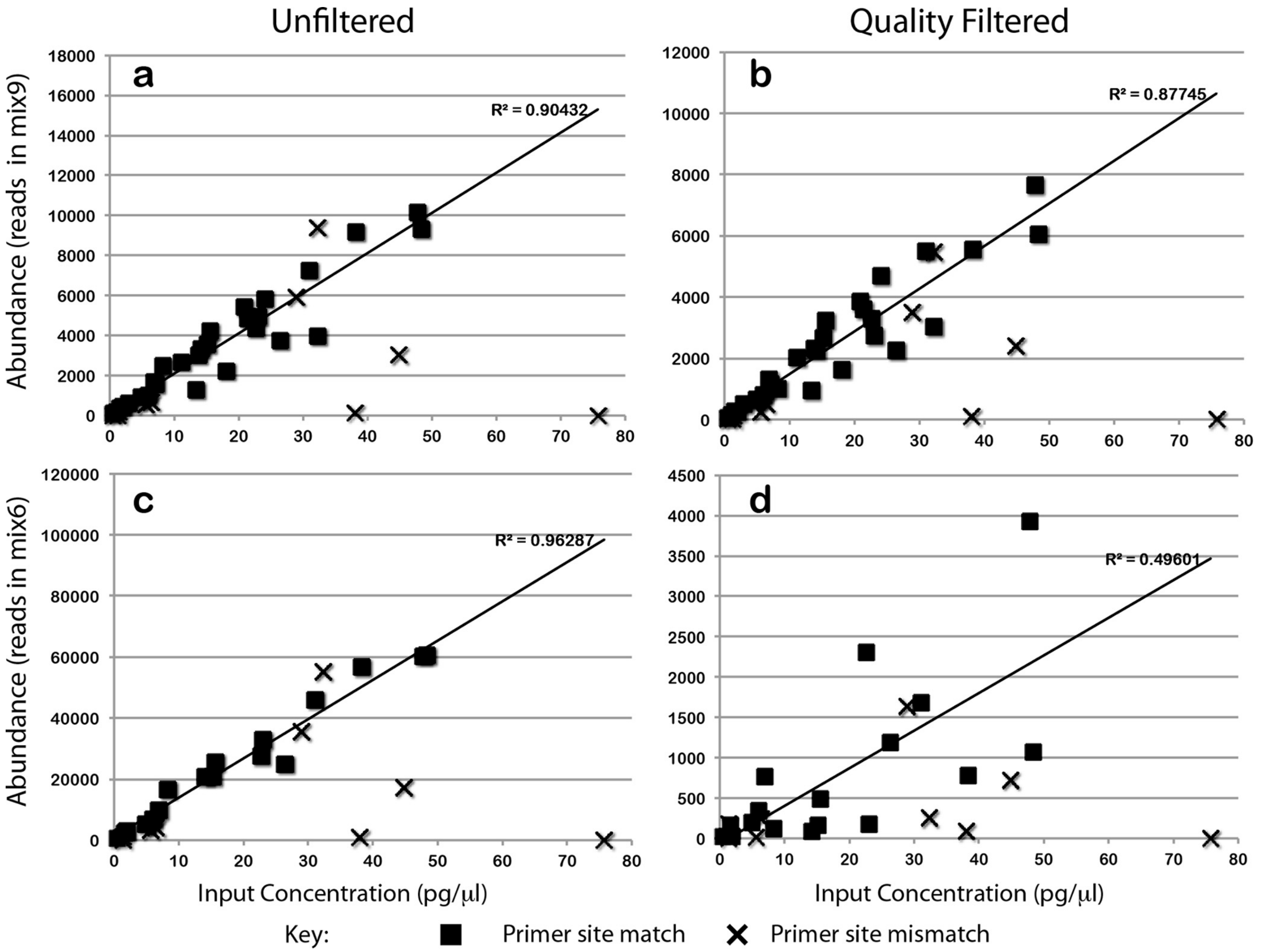

FIG 6 Template abundance is highly correlated with the input concentration when templates do not have mismatches in the primer-binding site. Additionally, stringent quality filtering can decrease the correlation with the input concentration for poor-quality sequencing runs. (a and b) Data from a high-quality sequencing run. (c and d) Data from a low-quality sequencing run. (a and c) Unfiltered data. (b and d) Filtered data. Abundance is determined as the number of reads with the best Blast hit to the mock-community sequence. Input concentrations were measured experimentally from the mock-community DNA template. Trend lines and corresponding correlation coefficients (R^2) are shown for reads with primer site matches only (black square).

number of pairwise comparisons and reduce the run time. However, they will also decrease the accuracy of the algorithm at removing incorrect OTUs (see Fig. S4 in the supplemental material).

There are some cases where the distribution-based clustering method should be used with caution. Distribution-based clustering predicts the most accurate OTUs when sequences are distributed in an ecologically meaningful way across samples, as in the mock community or in a stratified lake. However, methodological issues creating nonrandom errors across samples (e.g., different error rates across sequencing cells or runs) will increase the number of erroneous sequences that distribution-based clustering will keep as distinct OTUs (see Table S5 in the supplemental material). Nevertheless, distribution-based clustering still creates the most accurate OTUs of all clustering methods, even with the methodological errors found in the analysis. Users should also consider whether grouping sequences using a statistical test of similarity will impact the statistics of their downstream analyses.

Although no method formed OTUs that were as accurate as the distribution-based method with these mock communities, there are situations when different methods might be a more appropriate choice. Closed-reference clustering has the advantage of speed and convenience, especially for downstream processing, because information about the reference sequences can be precomputed (e.g., phylogenetic trees and taxonomic information). *De novo* clustering may be a good choice for higher-taxonomic-level analyses, as overclustering species should not affect phylum-level changes across samples, especially when the total number of predicted OTUs can affect the results. Open-reference clustering is less discriminating and tends to grossly overestimate the number of OTUs. However, it seems to be a good alternative when looking for trends between closely related organisms, especially if low-abundance OTUs can be filtered out.

When applied appropriately, each of the different clustering methods analyzed here can facilitate the discovery of important trends in 16S rRNA library sequence data. The introduction of the distribution-based clustering method gives researchers an additional tool that allows distinct OTUs to be retained even if they differ at a single base pair in a background of high microdiversity or sequencing error.

ACKNOWLEDGMENTS

This work, conducted by ENIGMA (Ecosystems and Networks Integrated with Genes and Molecular Assemblies) (http://enigma.lbl.gov), a Scientific Focus Area Program at Lawrence Berkeley National Laboratory, was supported by the Office of Science, Office of Biological and Environmental Research, of the U.S. Department of Energy under contract no. DE-AC02-05CH11231.

We thank Stuart Levin and the support staff at the MIT BioMicro Center for their help with Illumina sequencing. We thank John Casey for his help in improving the clarity of the manuscript.

REFERENCES

1. **Quince C, Lanzen A, Curtis TP, Davenport RJ, Hall N, Head IM, Read LF, Sloan WT.** 2009. Accurate determination of microbial diversity from 454 pyrosequencing data. Nat. Methods **6:**639–641.
2. **Huse SM, Welch DM, Morrison HG, Sogin ML.** 2010. Ironing out the wrinkles in the rare biosphere through improved OTU clustering. Environ. Microbiol. **12:**1889–1898.
3. **Quince C, Lanzen A, Davenport RJ, Turnbaugh PJ.** 2011. Removing noise from pyrosequenced amplicons. BMC Bioinformatics **12:**38.
4. **Schloss PD, Gevers D, Westcott SL.** 2011. Reducing the effects of PCR amplification and sequencing artifacts on 16S rRNA-based studies. PLoS One **6:**e27310. doi:10.1371/journal.pone.0027310.
5. **Edgar RC, Haas BJ, Clemente JC, Quince C, Knight R.** 2011. UCHIME improves sensitivity and speed of chimera detection. Bioinformatics **27:**2194–2200.
6. **Sun Y, Cai Y, Huse SM, Knight R, Farmerie WG, Wang X, Mai V.** 2012. A large-scale benchmark study of existing algorithms for taxonomy-independent microbial community analysis. Brief Bioinform. **13:**107–121.
7. **Sul WJ, Cole JR, Jesus Eda C, Wang Q, Farris RJ, Fish JA, Tiedje JM.** 2011. Bacterial community comparisons by taxonomy-supervised analysis independent of sequence alignment and clustering. Proc. Natl. Acad. Sci. U. S. A. **108:**14637–14642.
8. **Edgar RC.** 2010. Search and clustering orders of magnitude faster than BLAST. Bioinformatics **26:**2460–2461.
9. **Huang Y, Niu B, Gao Y, Fu L, Li W.** 2010. CD-HIT Suite: a web server for clustering and comparing biological sequences. Bioinformatics **26:**680–682.
10. **Zheng Z, Kramer S, Schmidt B.** 2012. DySC: software for greedy clustering of 16S rRNA reads. Bioinformatics **28:**2182–2183.
11. **Sun Y, Cai Y, Liu L, Yu F, Farrell ML, McKendree W, Farmerie W.** 2009. ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences. Nucleic Acids Res. **37:**e76. doi:10.1093/nar/gkp285.
12. **Schloss PD, Westcott SL.** 2011. Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. Appl. Environ. Microbiol. **77:**3219–3226.
13. **Caporaso JG, Lauber CL, Walters WA, Berg-Lyons D, Huntley J, Fierer N, Owens SM, Betley J, Fraser L, Bauer M, Gormley N, Gilbert JA, Smith G, Knight R.** 2012. Ultra-high-throughput microbial community analysis on the Illumina HiSeq and MiSeq platforms. ISME J. **6:**1621–1624.
14. **Hunt DE, David LA, Gevers D, Preheim SP, Alm EJ, Polz MF.** 2008. Resource partitioning and sympatric differentiation among closely related bacterioplankton. Science **320:**1081–1085.
15. **Koeppel A, Perry EB, Sikorski J, Krizanc D, Warner A, Ward DM, Rooney AP, Brambilla E, Connor N, Ratcliff RM, Nevo E, Cohan FM.** 2008. Identifying the fundamental units of bacterial diversity: a paradigm shift to incorporate ecology into bacterial systematics. Proc. Natl. Acad. Sci. U. S. A. **105:**2504–2509.
16. **Youngblut ND, Shade A, Read JS, McMahon KD, Whitaker RJ.** 2013. Lineage-specific responses of microbial communities to environmental change. Appl. Environ. Microbiol. **79:**39–47.
17. **Nemergut DR, Costello EK, Hamady M, Lozupone C, Jiang L, Schmidt SK, Fierer N, Townsend AR, Cleveland CC, Stanish L, Knight R.** 2011. Global patterns in the biogeography of bacterial taxa. Environ. Microbiol. **13:**135–144.
18. **Shapiro BJ, Friedman J, Cordero OX, Preheim SP, Timberlake SC, Szabo G, Polz MF, Alm EJ.** 2012. Population genomics of early events in the ecological differentiation of bacteria. Science **336:**48–51.
19. **Connor N, Sikorski J, Rooney AP, Kopac S, Koeppel AF, Burger A, Cole SG, Perry EB, Krizanc D, Field NC, Slaton M, Cohan FM.** 2010. Ecology of speciation in the genus Bacillus. Appl. Environ. Microbiol. **76:**1349–1358.
20. **Kunin V, Engelbrektson A, Ochman H, Hugenholtz P.** 2010. Wrinkles in the rare biosphere: pyrosequencing errors can lead to artificial inflation of diversity estimates. Environ. Microbiol. **12:**118–123.
21. **Degnan PH, Ochman H.** 2012. Illumina-based analysis of microbial community diversity. ISME J. **6:**183–194.
22. **Bokulich NA, Subramanian S, Faith JJ, Gevers D, Gordon JI, Knight R, Mills DA, Caporaso JG.** 2013. Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. Nat. Methods **10:**57–59.
23. **Turnbaugh PJ, Quince C, Faith JJ, McHardy AC, Yatsunenko T, Niazi F, Affourtit J, Egholm M, Henrissat B, Knight R, Gordon JI.** 2010. Organismal, genetic, and transcriptional variation in the deeply sequenced gut microbiomes of identical twins. Proc. Natl. Acad. Sci. U. S. A. **107:**7503–7508.
24. **Lane DJ.** 1991. 16S/23S rRNA sequencing, p 115–175. *In* Stackebrandt E, Goodfellow M (ed), Nucleic acid techniques in bacterial systematics. Wiley & Sons, Chichester, United Kingdom.
25. **Caporaso JG, Lauber CL, Walters WA, Berg-Lyons D, Lozupone CA, Turnbaugh PJ, Fierer N, Knight R.** 2011. Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. Proc. Natl. Acad. Sci. U. S. A. **108:**4516–4522.
26. **Blackburn MC.** 2010. Development of new tools and applications for high-throughput sequencing of microbiomes in environmental or clinical samples. M.Sc. thesis. Massachusetts Institute of Technology, Cambridge, MA.
27. **Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer N, Pena AG, Goodrich JK, Gordon JI, Huttley GA, Kelley ST, Knights D, Koenig JE, Ley RE, Lozupone CA, McDonald D, Muegge BD, Pirrung M, Reeder J, Sevinsky JR, Turnbaugh PJ, Walters WA, Widmann J, Yatsunenko T, Zaneveld J, Knight R.** 2010. QIIME allows analysis of high-throughput community sequencing data. Nat. Methods **7:**335–336.
28. **Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ, Sahl JW, Stres B, Thallinger GG, Van Horn DJ, Weber CF.** 2009. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl. Environ. Microbiol. **75:**7537–7541.
29. **Price MN, Dehal PS, Arkin AP.** 2010. FastTree 2—approximately maximum-likelihood trees for large alignments. PLoS One **5:**e9490. doi:10.1371/journal.pone.0009490.
30. **Baldi P, Brunak S, Chauvin Y, Andersen CAF, Nielsen H.** 2000. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics **16:**412–424.
31. **Polz MF, Cavanaugh CM.** 1998. Bias in template-to-product ratios in multitemplate PCR. Appl. Environ. Microbiol. **64:**3724–3730.
32. **Lahr DJ, Katz LA.** 2009. Reducing the impact of PCR-mediated recombination in molecular evolution and environmental studies using a new-generation high-fidelity DNA polymerase. Biotechniques **47:**857–866.
33. **Dohm JC, Lottaz C, Borodina T, Himmelbauer H.** 2008. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. Nucleic Acids Res. **36:**e105. doi:10.1093/nar/gkn425.
34. **Nakamura K, Oshima T, Morimoto T, Ikeda S, Yoshikawa H, Shiwa Y, Ishikawa S, Linak MC, Hirai A, Takahashi H, Altaf-Ul-Amin M, Ogasawara N, Kanaya S.** 2011. Sequence-specific error profile of Illumina sequencers. Nucleic Acids Res. **39:**e90. doi:10.1093/nar/gkr344.
35. **Minoche AE, Dohm JC, Himmelbauer H.** 2011. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. Genome Biol. **12:**R112.
36. **Papa E, Docktor M, Smillie C, Weber S, Preheim SP, Gevers D, Giannoukos G, Ciulla D, Tabbaa D, Ingram J, Schauer DB, Ward DV, Korzenik JR, Xavier RJ, Bousvaros A, Alm EJ.** 2012. Non-invasive mapping of the gastrointestinal microbiota identifies children with inflammatory bowel disease. PLoS One **7:**e39242.
37. **Ley RE, Backhed F, Turnbaugh P, Lozupone CA, Knight RD, Gordon JI.** 2005. Obesity alters gut microbial ecology. Proc. Natl. Acad. Sci. U. S. A. **102:**11070–11075.