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

Citation: Preheim, S. P., A. R. Perrotta, A. M. Martin-Platero, A. Gupta, and E. J. Alm. "Distribution-Based Clustering: Using Ecology To Refine the Operational Taxonomic Unit." Applied and Environmental Microbiology 79, no. 21 (August 23, 2013): 6593–6603.

As Published: http://dx.doi.org/10.1128/AEM.00342-13

Publisher: American Society for Microbiology

Persistent URL: http://hdl.handle.net/1721.1/91469

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

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

Running title: Using Ecology to Refine the Operational Taxonomic Unit

Sarah P. Preheim[1#], Allison R. Perrotta[2], Antonio M. Martin-Platero[2], Anika Gupta[2] and Eric J. Alm[1#]

[1] Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, MA; [2] Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA

[#]Corresponding author email: ejalm@mit.edu, spacocha@mit.edu

Abstract

16S ribosomal RNA (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 identify taxonomic groups or eliminate 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 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. This program can decrease the total number of OTUs with redundant information and improve the power of many down-stream analyses to describe biologically relevant trends.

Introduction

Identifying meaningful OTUs is a significant bottleneck in the analysis of 16S rRNA sequences from complex microbial communities, particularly for large datasets generated by next-generation sequencing. Spurious sequences created by PCR or sequencing errors can greatly inflate the total number of OTUs (i.e. 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 cut-off 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 cut-off. Many heuristics have been developed to decrease computational demand of OTU calling with varying degrees of accuracy, such as CD-HIT (9), UCLUST (8), DySC (10) and ESPRIT (11), among others. 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 cut-off, typically >97% sequence identity, to inform OTU clustering.

Although it is common to use a single sequence identity cut-off for clustering, more insight can be gained by adjusting the sequence clustering for individual taxonomic lineages (14, 15) or by using multiple genetic cut-offs for analysis (16, 17). Hunt *et al* (14) developed a program called AdaptML to infer population boundaries from the ecological information of isolated strains. Different populations were often identified within what would generally be considered one species. Using two closely related populations predicted from AdaptML, Shapiro *et al* (18) was 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 is supported by AdaptML using ecology alone (19). Both Youngblut *et al* (16) and Nemergut *et al* (17) repeated their analysis at various levels of clustering. Youngblut *et al* (16) found that using an inappropriate genetic cut-off 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 pyrosequencing (1, 3-5, 20) and Illumina datasets

(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 dataset, reducing the effective error rate. However, these methods do not help in identifying how these "cleaned" sequences should be grouped into OTUs for down-stream 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 a different distribution 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, variation of the 16S rRNA sequence within a population or sequences generated randomly in error. Thus, whether the underlying distribution is the same for ecological (i.e. same population of bacteria) or methodological reasons (i.e. sequencing error), they should be considered as 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 from 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 abundance of sequences within all samples and genetic distance to inform clustering. We compare this method ("distribution-based clustering" or DBC) to commonly applied closed-reference (i.e. phylotyping), open-reference (i.e. hybrid of phylotyping and *de novo* clustering) and *de novo* clustering methods using experimental mock community datasets. 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 it calls fewer OTUs than either the *de novo* or open-reference methods, 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 dataset that was previously generated (23) to test our clustering method. Data was downloaded from the Supplemental Data page from the Gordon Lab website for the paper (http://gordonlab.wustl.edu/TurnbaughSE_2_10/PNAS_2010.html). The quality filtered, denoised and chimera-free dataset was used for further analysis (Mock_nochimeras.fna) and all sequences were trimmed to 210 bases and the first 14 bases were removed. The input sequences (MockIsolatesV2.fna) and the input distributions from the Supplemental Material [Table S3 in Turnbaugh *et al* (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 dataset (Mock_cleaned.fna).

The representative sequence for *Providencia alcalifacien* was mislabeled as *Providencia rettgeri*, as evident from distribution of this sequence across samples [which corresponded to the *Providencia alcalifacien* distribution (Fig. S1a)] and matched many *Providencia alcalifacien* strains in NCBI's nr database. The *Providencia rettgeri* sequence was replaced with the sequence from the dataset that had the correct corresponding distribution (Fig. S1b) and that matched many *Providencia rettgeri* sequences in NCBI's 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. DNA templates were 16S rRNA sequences on purified, linearized plasmids (i.e.

151 environmental clones) as described in the Supplementary Information (SI) and
152 approximately 800 base pairs were sequenced from the forward primer 27F (24).
153 All clone sequences were submitted to GenBank (accession nos. KC192376 -
154 KC192544). The input concentration of each DNA template was measured using the
155 2100 Bioanalyzer (Agilent Technologies Inc., Santa Clara, CA). DNA templates were
156 mixed together into nine different mock communities ranging from simple (com1)
157 with five DNA templates added to complex (com9) with 40 total DNA templates.
158 DNA templates were mixed to create a range of final concentrations. Specific
159 information about mock community composition can be found in Table S1 and S2.
160
161 *Library construction and sequencing*
162 Mock community libraries for paired-end Illumina sequencing were
163 constructed using a two-step 16S rRNA PCR amplicon approach diagramed in Figure
164 S2. The first step primers (PE16S_V4_U515_F : 5' ACACG ACGCT CTTCC GATCT
165 YRYRG TGCCA GCMGC CGCGG TAA- 3'; PE16S_V4_E786_R: 5'- CGGCA TTCCT GCTGA
166 ACCGC TCTTC CGATC TGGAC TACHV GGGTW TCTAA T 3') contain primers U515F
167 and E786R targeting the V4 region of the 16S rRNA gene as described previously
168 (25). Additionally, a complexity region in the forward primer (5'-YRYR-3') was
169 added to aid the image processing software used detect distinct clusters during
170 Illumina next generation sequencing. A second-step priming site is also present in
171 both the forward (5'-ACACG ACGCT CTTCC GATCT-3') and reverse (5'-CGGCA
172 TTCCT GCTGA ACCGC TCTTC CGATC T-3') first step primers. The second step
173 primers incorporate the Illumina adapter sequences and a nine base pair 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's indicate the presence of a unique barcode listed in Table S3).

Real-time PCR before the first-step PCR reaction was done to ensure uniform amplification and avoid over-cycling all templates. Both real-time and first step PCR reactions were done similar to the manufacture's protocol for Phusion polymerase (New England Biolabs, Ipswich, MA) as described in SI. 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). 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 machines at the BioMicro Center (MIT, Cambridge, MA). Environmental libraries were sequenced with 2 x 144 bases and mock community samples done with 2 x 100.

Pre-and post-clustering quality control

Raw data was 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 a base quality dropped below Phred quality score 17, which corresponds to a probability of error

197 around 0.02 (--last_bad_character Q -r 0). This quality filter stringency was chosen
198 because it was found to result in the smallest Jensen-Shannon divergence from the
199 true distribution using com9 (Fig. S3). Only sequences at least 99 bps long after
200 quality filtering were retained (-min_per_read_length 99). All other parameters
201 were default parameters. After quality filtering, the complexity region between the
202 adapters and the primer (Fig. S2), along with the primer sequence were removed
203 using the trim.seqs program in mothur (version v.1.23.1) (28) and trimmed to 76 bp
204 with a custom perl script (https://github.com/spacocha/Distribution-based-
205 clustering/blob/master/bin/truncate_fasta.pl). All sequences not matching the first
206 15 bases of the primer were removed.
207 After each clustering algorithm, representative sequences were picked using
208 QIIME pick_rep_set.py, or a custom perl script
209 (https://github.com/spacocha/Distribution-based-
210 clustering/blob/master/bin/pick_most_ab_from_ablist.pl), using the most abundant
211 sequence in the OTU as the representative. These were used to determine which
212 OTUs were correct (i.e. matching an input sequence) or incorrect (i.e. did not match
213 an input sequence). OTUs were removed if the representative sequence did not align
214 to the part of the 16S rRNA gene that was amplified (positions 13862 to 15958 of
215 the silva alignment) with at least 76 bp. OTUs with less than 2 counts, or 11 counts
216 were filtered out using QIIME's filter_otu_table.py (-c 2 or -c 11) for data in Table 2.
217
218 *Closed-reference, open-reference and* de novo *clustering methods*

QIIME was used to make closed-reference (i.e. phylotype) and open-reference (i.e. hybrid of phylotyping and *de novo* approaches) OTUs as described above. Closed-, and open-reference clustering was done with the pick_reference_otus_through_otu_table.py flow from QIIME. Both methods used the 12_10 greengenes 97% reference OTU collection (http://qiime.org/home_static/dataFiles.html) as the reference, 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 SI.

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 SI. The total number of OTUs was similar after chimera checking and lineage removal.

USEARCH (v. 6.0.307; http://www.drive5.com/usearch/) was used to create the USEARCH *de novo* OTU with custom perl scripts for pre- and post- processing as described in SI, 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 one base but that are found in different samples, suggesting they did not arise from the same underlying distribution, and 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, 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 as a unit (Fig. 1b). A statistical test (i.e. chi-squared test) can be used to determine whether two sequences have a similar distribution across libraries. Applying these metrics can merge sequences derived from the same population (e.g. sequencing error or inter-operon 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 non-random way across samples (i.e. higher error rates in a subset of libraries).

Distribution-based clustering algorithm

Distribution-based clustering requires two input files, an OTU-by-library matrix and a distance 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 step-wise manner (Fig. 2), as described below.

Step 1.) 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.

Step 2.) 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 distance, to reduce the impact of misalignment. If an OTU is found whose representative sequence is within the genetic distance cut-off, proceed to step 3. Otherwise, stop the search and go to step 6.

Step 3.) 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), 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 for comparisons with candidate sequences that are at a similar abundance to the OTU representative (-abund 0). This low abundance threshold provides the possibility to merge sequences together that were generated from inter-operon 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.

Step 4.) Determine whether the candidate and representative sequences are distributed across samples in a similar manner. 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 distribution of candidate sequence (i.e.

observed distribution) and the OTU representative sequence (i.e. expected distribution) is similar if the chi-squared test results in a p-value above a user-defined cut-off (default = 0.0005). Sequences with low counts (e.g. singletons) will also result in high p-values. P-values are calculated using R statistical language (chisq.test) or simulated (chisq.test:simulate.p.value) when the expected is below 5 for more than 80% of the compared values. As an additional option, the Jensen-Shannon divergence (JSD) can be used. 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). 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.

Step 5) Add the sequence to 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.

Step 6.) 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 sequences OTUs (Fig. S4). Default parameters were used to cluster the mock community sequences generated in this study. The previously generated, cleaned dataset (23) was clustered with the following parameters: distance cut-off was 0.05, abundance criteria 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 vs. Parallel algorithm

With the "complete" process, all sequences were analyzed together in the analysis. In the "parallel" process, sequences were pre-clustered with a heuristic approach (see below) and sequences in each cluster were processed separately, in parallel. However, sequences could be pre-clustered with different algorithms (e.g. nearest-neighbor single linkage clustering), as long as the number of sequences that are grouped with their nearest neighbor is maximized. Data was pre-clustered with UCLUST into clusters for the new mock 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 generated in this paper, and 0.95 for the Turnbaugh mock community. 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. 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 one base pair of difference between them. The corresponding input organisms or template for each resulting sequence was determined as the smallest distance (minimum of aligned and unaligned distances) to an input sequence for each unique sequence. Sequences were weighted by abundance, so more abundant sequences result in more total counts.

To assess the accuracy of each method against our criteria, we used two measures of a test's accuracy, F-score and 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

379 sequences in different OTUs originating from different inputs. False negatives (FN)
380 are defined as a pair of sequences in different OTUs originating from the same input
381 or if either of a pair of reads was not assigned into an OTU (only affecting closed-
382 reference clustering). These were calculated with various scripts using the resulting
383 OTU list from each algorithm along with a mapping file indicating the input
384 (determined as above) and a translation file mapping reads to libraries
385 (https://github.com/spacocha/Distribution-based-
386 clustering/tree/master/confusion_matrix_calc).

387 The F-score was calculated as:

388 $$f\text{-}score = 2 \times \frac{precision \times recall}{precision + recall}$$

389 where precision is defined as:

390 $$precision = \frac{TP}{TP + FP}$$

391 and recall is defined as:

392 $$recall = \frac{TP}{TP + FN}$$

393 Matthew's correlation coefficient (MCC):

394 $$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}}$$

395 as previously described (30) with the definition of TP, FP, TN and FN described
396 above.

397

398 *Comparison with the input community*

To compare the resulting OTU by library matrix with the expected distribution [Table S3 of Turnbaugh *et al*. (23) and Table S2 in SI], we used the JSD from mock community com9 and Uneven2 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). JSD was calculated using dist_mat (metric='JS') using PySurvey (https://bitbucket.org/yonatanf/pysurvey).

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 keep separate those originating from different input organisms or templates (Fig. 1). Sequences derived from the same input could represent micro-diversity from inter-operon 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

422 different experimental mock communities to demonstrate how this algorithm
423 compares to more commonly applied clustering methods based on sequence
424 identity alone.
425
426 *Distribution-based clustering more accurately clusters sequences created in error*
427 Distribution-based clustering creates OTUs that more accurately represent
428 the input sequences based on the total number of OTUs, how sequences are grouped
429 together into OTUs and distribution of OTUs across samples. 38 mock template
430 sequences remain in distinct OTUs in both distribution-based and open-reference
431 clustering, resulting in the largest number of OTUs containing at least one of the
432 input sequences ("Correct"; Fig. 3a). *De novo* clustering has fewer correct OTUs
433 because some sequences are merged into the same OTU. Closed-reference clustering
434 retains fewer correct OTUs because some of the community members do not match
435 the database with sufficient identity. Distribution-based clustering predicts the
436 lowest number of spurious, incorrect OTUs ("Incorrect"; Fig. 3a). Open-reference
437 clustering predicts the largest number of incorrect OTUs of all methods.
438 Distribution-based clustering also groups together reads originating from the
439 same template sequence more accurately. A typical benchmark of OTU accuracy is
440 whether the algorithms cluster sequences that are within a specific genetic distance
441 or sequence identity threshold (12). However, our benchmark is whether reads that
442 originate from the same mock template are grouped together and reads originating
443 from different templates are kept apart. The F-score and Matthew's correlation
444 coefficient are both measures of classification accuracy that have been used

445 previously to benchmark OTU definitions (12). By either metric, distribution-based
446 clustering out-performs all of the other methods at accurately discriminating input
447 sequences (Fig. 3b). *De novo* clustering predicts more true positives than
448 distribution-based clustering, but also predicts about 10 times more false positives
449 than distribution-based clustering (Table 1) because it tends to over-cluster the
450 closely related true sequences. Closed-reference clustering has the lowest scores
451 due to a large number of false negatives for sequences that do not match the
452 database.

453 Distribution-based clustering produces a resulting community that is more
454 similar to the input community in both total number and relative abundance of
455 OTUs. The number of reads mapping to each OTU from one high quality library
456 (com9) was compared to the input sequences using the Jensen-Shannon divergence
457 (Fig. 3c). Distribution-based clustering (both complete and parallel applications- see
458 *Complete vs. Parallel algorithm* in Materials and Methods for details) had the
459 smallest Jensen-Shannon divergence from the input community of all clustering
460 algorithms. Both *de novo* algorithms result in the largest divergence from the true
461 distribution of all clustering methods because some input sequences are merged
462 together. Closed-reference clustering discarded many input sequences that did not
463 match the database, resulting in a larger calculated divergence from the input
464 community. Open-reference clustering does not merge as many input sequences as
465 *de novo* clustering and does not discard any true sequences like closed-reference
466 clustering, but was still less accurate than distribution-based clustering.
467

Filtering out low abundance OTUs

Low abundance OTUs are often discarded because they do not contain much information. We have also compared the total number 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 in the environmental sample. However, the total number of OTUs is 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 was previously cleaned, 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 sequences together 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 fewest 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. under-clustering). 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 sequence (i.e. over-clustering). These results highlight the drawback of using genetic information alone, which will necessarily either over-cluster or under-cluster 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 input when the level of genetic variation is different across taxonomic lineages.

Comparison with unknown samples

Along with comparisons between clustering methods on 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 datasets for this analysis. First, we made an Illumina 16S rRNA library from

514 DNA extracted from water collected approximately every meter from the surface to
515 the bottom (22 meters depth). Additionally, we generated Sanger-sequencing based
516 16S rRNA libraries from two depths (6 and 21 meters). The distribution of the
517 Illumina library sequences was used in the clustering method and Illumina
518 sequences that match different Sanger clones were used as a control comparison,
519 since these sequences were observed independently in the Illumina and Sanger
520 datasets.
521
522 Closed-reference clustering over-filters environmental data
523 The closed-reference clustering method predicts the fewest number of OTUs
524 of all methods (Fig. 5a). Although the total number of OTUs in the sample is
525 unknown, the Illumina sequences that match the Sanger library mock community
526 can be used to compare clustering methods on the unknown sample. 89 Illumina
527 sequences match one or more of the Sanger sequences. As we saw with the simple
528 mock community, which was generated from clones of these sequences, the closed-
529 reference method discards many sequences that are missing representative
530 sequences in the database. Closed-reference clustering discards 15 of the 89
531 sequences with more than 1000 counts across all libraries. The most abundant
532 discarded sequence is classified as Cyanobacteria with a distribution that
533 corresponds to a peak in oxygen below the thermocline. This suggests that the very
534 low number of OTUs predicted by the closed-reference method is an underestimate
535 and that this method excluded biologically interesting information.
536

Over-clustered environmental data

De novo, open-, and closed- reference clustering over-clustered the data, resulting in skewed environmental distributions for many OTUs as compared with distribution-based clustering. Merged sequences with different distributions resulted low correlations between the resulting OTU and the matching Sanger clone for different clustering methods because merged sequences had very distinct profiles (such as in Fig. 5b). The distribution of five OTUs formed by *de novo* (USEARCH) clustering, resulting in correlations below 0.9 with the matching Sanger sequence (Table S4). Three OTUs formed by open- and closed-reference clustering algorithms had low correlations with the matching Sanger sequence (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 over cluster sequences with distinct environmental distributions, as compared to distribution-based clustering.

Distribution-based clustering is accurate and flexible

The 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* open-reference (Fig 5a). When filtering out singletons (i.e. OTUs with 1 count), distribution-based clustering still predicted a few 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 original sequence. This is comparable to the analysis done with the mock community generated from 16S plasmid templates (Fig. 3). Under these conditions, sequences in Fig. 5c would remain as distinct OTUs. However, it may be redundant to consider each sequence as a separate OTU because they are genetically similar and distributed in a similar manner. Thus, the distribution-based algorithm can also be adjusted to merge sequences in Fig. 5c by using no abundance cut-off and comparing the sequence distributions with 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 predicts 11,871 OTUs total, 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 computational time for distribution-based clustering is much longer than any of the other clustering methods. Table 3 shows typical run times for

583 approximately 500,000 total reads (5,489 unique sequences) in the mock and 7.5
584 million reads (120,601 unique sequences). Only the parallelized distribution-based
585 clustering used multiple processors to complete, and run time of other methods
586 could be improved even further by using multiple processors. However, it is clear
587 that there is a significant difference in speed between distribution-based clustering
588 and the other methods.
589
590 *Issues affecting sequence and distribution accuracy*
591 The sequences and distribution of OTUs across libraries should represent the
592 true distribution as accurately as possible. Recommendations made from previous
593 studies were followed during library construction to reduce PCR amplification
594 biases, including reducing the cycle number and pooling replicate PCR reactions (31,
595 32). While these measures help, the resulting sequences and distributions across
596 libraries is primarily affected by two things: mismatches between the primer and
597 template sequences and sequence-specific errors of the Illumina sequencing
598 platform from a poor quality run.
599
600 <u>Sequence-specific sequencing errors</u>
601 The distribution-based clustering method is sensitive to errors that are
602 generated in a non-random way across samples. Since the algorithm assumes that
603 differences in the distribution of sequences across samples represent important
604 information, this assumption is invalid when differences are due to methodological
605 errors. In our analysis, the most obvious cause of non-random errors results from

606 combining sequencing data from different runs with varying quality scores (Fig. S5),
607 as certain errors were generated at a higher frequency on one flow-cell than the
608 other (Fig. S6a). This causes the erroneous sequences to have a significantly
609 different distribution than the sequences they were derived from (Fig. S6b), and are
610 thus retained as distinct OTUs. As expected, distribution-based clustering performs
611 very well on simulated data when the error rate is constant across libraries, but is
612 substantially worse when error rates are non-constant (Table S5). Thus,
613 distribution-based clustering would be even more accurate had all of the samples
614 been sequenced on the same flow-cell.
615 Sequence specific errors are obvious when a stringent quality filter is applied
616 to a low-quality sequencing lane. After removing templates with primer site
617 mismatches, Fig. 6 shows little decrease in the correlation between the observed
618 and expected frequencies for a good quality sequencing run after quality filtering
619 (Fig. 6a and 6b). In a library from the poor quality lane (Flow 2 lane 1, com4-com6),
620 the correlation with the input concentration is high for unfiltered data (R^2=0.96287;
621 Fig. 6c). However, the correlation between the input concentration and the resulting
622 sequences breaks down with more stringent quality filtering (R^2=0.49601; Fig. 6d).
623 This is likely due to sequence specific errors, a problem identified previously with
624 Illumina sequencing technology (33-35). When using data from poor quality
625 sequencing runs, OTUs from more stringent quality filtering represent true
626 sequences, but the relative abundances may be highly skewed.
627
628 **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 coordinates analysis (PCoA) plots, which identify the most obvious differences between samples, were similar across clustering methods (Fig. S7 and S8). PCoA are particularly effective when the variable of interest (e.g. depth, disease state) is associated with major changes in community structure, but are less effective at detecting subtle variations in community structure. Furthermore, they 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

651 clustering may be one of only a few tools that can be used to distinguish the signal
652 from the noise of sequencing errors.

653 Run time is currently a severe limitation to implementing distribution-based
654 clustering on very large datasets. Although many improvements can be made to the
655 algorithm itself to increase the speed of the program (likely with lower accuracy),
656 any implementation will likely be more computationally intensive than other
657 methods since it involves processing additional information. Steps can be taken to
658 reduce total runtime, such as increasing the abundance skew (e.g. 100-fold more
659 abundant representative sequences) or decreasing the total distance cut-off allowed
660 for forming clusters (e.g. cutoff=0.05), or filtering out low abundance sequences (e.g.
661 filter out singletons). All of these steps decrease the total number of pairwise
662 comparisons and reduce runtime. However, these will also decrease accuracy of the
663 algorithm at removing incorrect OTUs (Fig. S4).

664 There are some cases when the distribution-based clustering method should
665 be used with caution. Distribution-based clustering predicts the most accurate OTUs
666 when sequences are distributed in an ecologically meaningful way across samples,
667 as in the mock community or in a stratified lake. However, methodological issues
668 creating non-random errors across samples (e.g. different error rates across
669 sequencing cells or runs) will increase the number of erroneous sequences that
670 distribution-based clustering will keep as distinct OTUs (Table S5). However,
671 distribution-based clustering still creates the most accurate OTUs of all clustering
672 methods, even with the methodological errors found in this analysis. Users should

673 also consider whether grouping sequences using a statistical test of similarity will
674 impact the statistics of their downstream analyses.

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

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

693

694 **Acknowledgements**

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 MIT's BioMicro Center for their help with Illumina sequencing. We thank John Casey for his help 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. Environmental Microbiology **12:**1889-1898.
3. **Quince C, Lanzen A, Davenport RJ, Turnbaugh PJ.** 2011. Removing Noise From Pyrosequenced Amplicons. BMC Bioinformatics **12**.
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**.
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. Proceedings of the National Academy of Sciences of the United States of America **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.

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.
12. **Schloss PD, Westcott SL.** 2011. Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. Applied and Environmental Microbiology **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. The ISME journal **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. Proceedings of the National Academy of Sciences **105:**2504-2509.
16. **Youngblut ND, Shade A, Read JS, McMahon KD, Whitaker RJ.** 2013. Lineage-Specific Responses of Microbial Communities to Environmental Change. Applied and Environmental Microbiology **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. Environmental Microbiology **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. Applied and Environmental Microbiology **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. Environmental Microbiology **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-U11.
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. Proceedings of the National Academy of Sciences of the United States of America **107:**7503-7508.

24. **Lane DJ.** 1991. 16S/23S rRNA sequencing, p. 115-175. *In* Stackebrandt E, Goodfellow M (ed.), Nucleic Acid Techniuqes in Bacterial Systemantics. Wiley & Sons, Chinchester.
25. **Knight R, Caporaso JG, Lauber CL, Walters WA, Berg-Lyons D, Lozupone CA, Turnbaugh PJ, Fierer N.** 2011. Global patterns of 16S rRNA diversity at a depth of millions of sequences per sample. Proceedings of the National Academy of Sciences of the United States of America **108:**4516-4522.
26. **Blackburn MC.** 2010. Development of New Tools and Applications for High-Throughput Sequencing of Microbiomes in Environmental or Clinical Samples. Master of Science in Chemical Engineering. 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. Applied and Environmental Microbiology **75:**7537-7541.
29. **Price MN, Dehal PS, Arkin AP.** 2010. FastTree 2-Approximately Maximum-Likelihood Trees for Large Alignments. Plos One **5**.
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. Applied and Environmental Microbiology **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.
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.
35. **Minoche AE, Dohm JC, Himmelbauer H.** 2011. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. Genome Biology **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**.

37. **Ley RE, Backhed F, Turnbaugh P, Lozupone CA, Knight RD, Gordon JI.** 2005. Obesity alters gut microbial ecology. Proceedings of the National Academy of Sciences of the United States of America **102:**11070-11075.

Titles and Legends to Figures

Figure 1 Schematic of how the distribution-based clustering algorithm forms OTUs. Symbols represent sequences originating from the same template, organism or population. Gray coloring represent dominant sequences, and white coloring represent lowly abundant variants or errors. OTUs are represented as ovals encompassing one or more symbols. (a) Hypothetical phylogenetic tree of the genetic relationship between various sequences represented by different symbols and colors. 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) The phylogenetic relationship and distribution of a dominant sequence and a lowly abundant 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 percent sequence identity cut-off will keep all dominant sequences in separate OTUs, but will keep some lowly abundant or erroneous sequences in different OTUs. Alternatively, using a lower identity cut-off, all lowly abundant variants will be merged with the abundant variants, but the diamond and square symbols are merged as well.

Figure 2 An outline of the decision making process used during distribution-based clustering. Rounded rectangles indicate the beginning and end of the process and arrow point to the next step in the process. Hexagons indicate a loop with the sorting criteria indicated within the hexagon. Diamonds indicate a decision step, with the question contained within the diamond and arrows directing how the program will respond. Rectangles indicate action steps, where sequences are categorized as either representatives of a new OTU or merged into an existing OTU.

Figure 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.) The number of total correct (black- left axis) and incorrect (hatched- right axis) OTUs predicted by each clustering method. A correct OTUs is one in which the representative sequence matches one of the input sequences. b.) The accuracy of each clustering method at grouping together reads originating from the same template as measured by both F-score (black bar) and Matthew's coefficient correlation (hatched bar). c.) The Jensen-Shannon divergence (JSD) is used as a measure of distance from the input of resulting communities created by applying each clustering method.

Figure 4 Distribution-based clustering predicts fewer OTUs and more accurately clustered reads originating from the same organism in cleaned, denoised and chimera-free mock community. a.) The number of total OTUs predicted by each clustering method. b.) The accuracy of each clustering method at grouping together

878 sequences that originated from the same organism as measured by both F-score
879 (black bar) and Matthew's coefficient correlation (hatched bar). c.) The Jensen-
880 Shannon divergence (JSD) is used as a measure of distance from the input of
881 resulting communities created by applying each clustering method.
882
883 **Figure 5** Evaluation of clustering methods on environmental samples from a
884 stratified lake. a.) The total number of OTUs predicted by each clustering method for
885 the entire lake. b.) Sequences displaying distinct ecological distributions but are
886 merged by all clustering algorithms except distribution-based clustering. The solid
887 line is the distribution of the resulting cluster, comprised mainly of two sequences
888 (dotted and dashed lines). Distribution-based clustering keeps the two sequences
889 distinct, but all other methods merge them into one OTU. c.) Sequences that
890 represent micro-diversity within the environmental sample. The distribution-based
891 clustering algorithm can be adjusted such that these sequences either remain
892 distinct or can be clustered.
893
894 **Figure 6** Template abundance is highly correlated with input concentration when
895 templates do not have mismatches in the primer-binding site. Additionally, stringent
896 quality filtering can decrease the correlation with input concentration for poor
897 quality sequencing runs. a) and b) show data from a high quality sequencing run and
898 c.) and d.) show data from a low quality sequencing run. a) and c) show unfiltered
899 data and b.) and d.) show filtered data. Abundance is determined as the number of
900 reads with best blast hit to the mock community sequence. Input concentrations

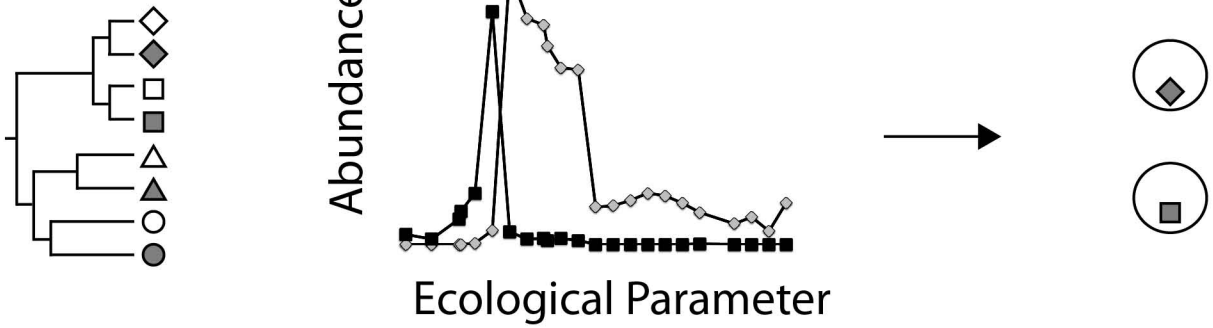
901 were measured experimentally from mock community DNA template. Trendlines
902 and corresponding correlation coefficients (R^2) shown for reads with primer site
903 matches only (black square).

a

Sequence Data

Distribution Information

Desired Clustering

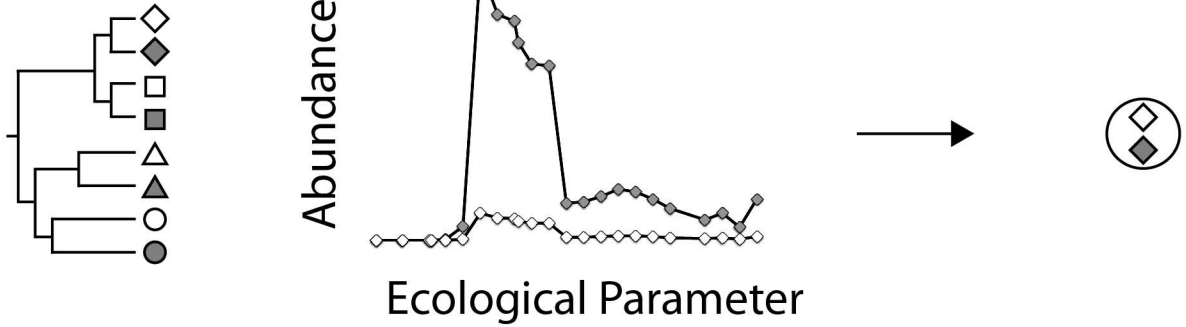


b

Sequence Data

Distribution Information

Desired Clustering

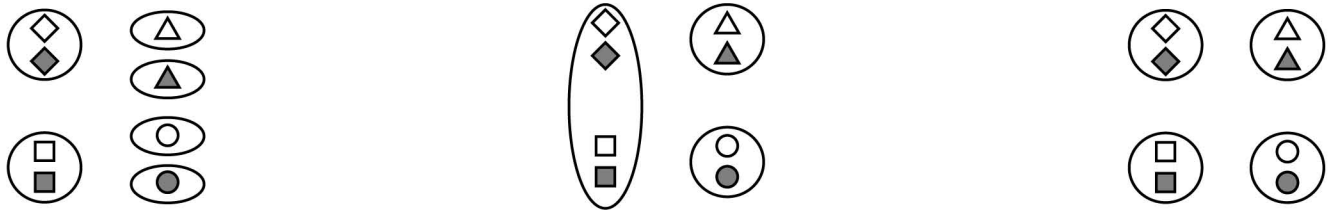


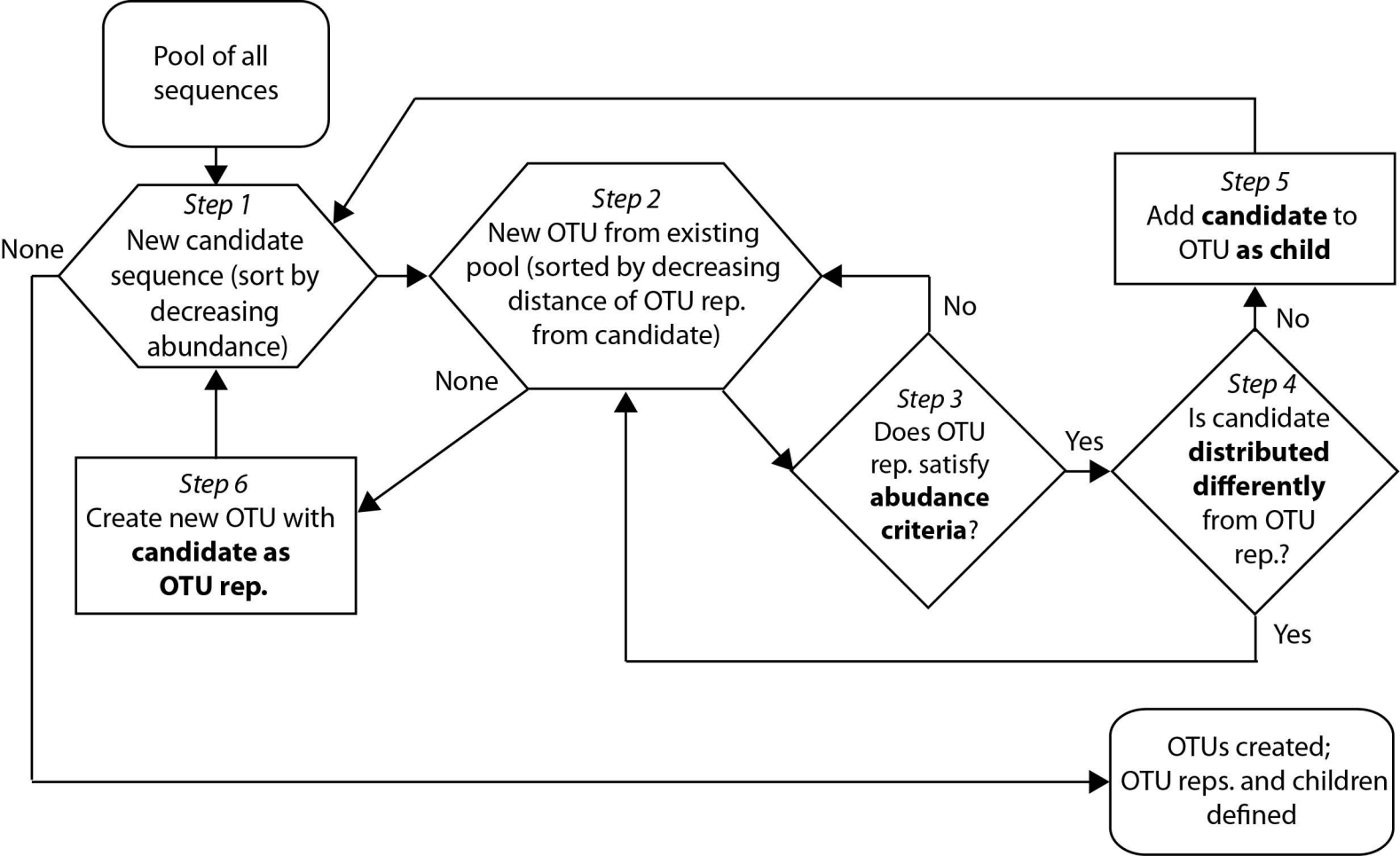
c

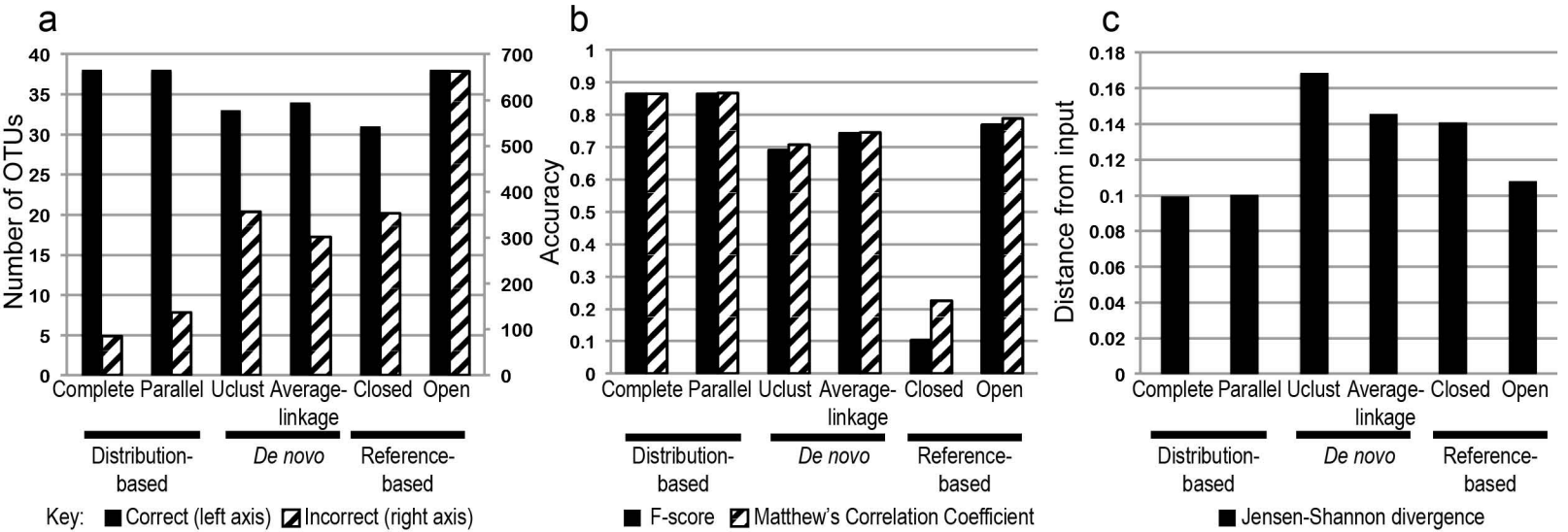
Higher Identity Threshold

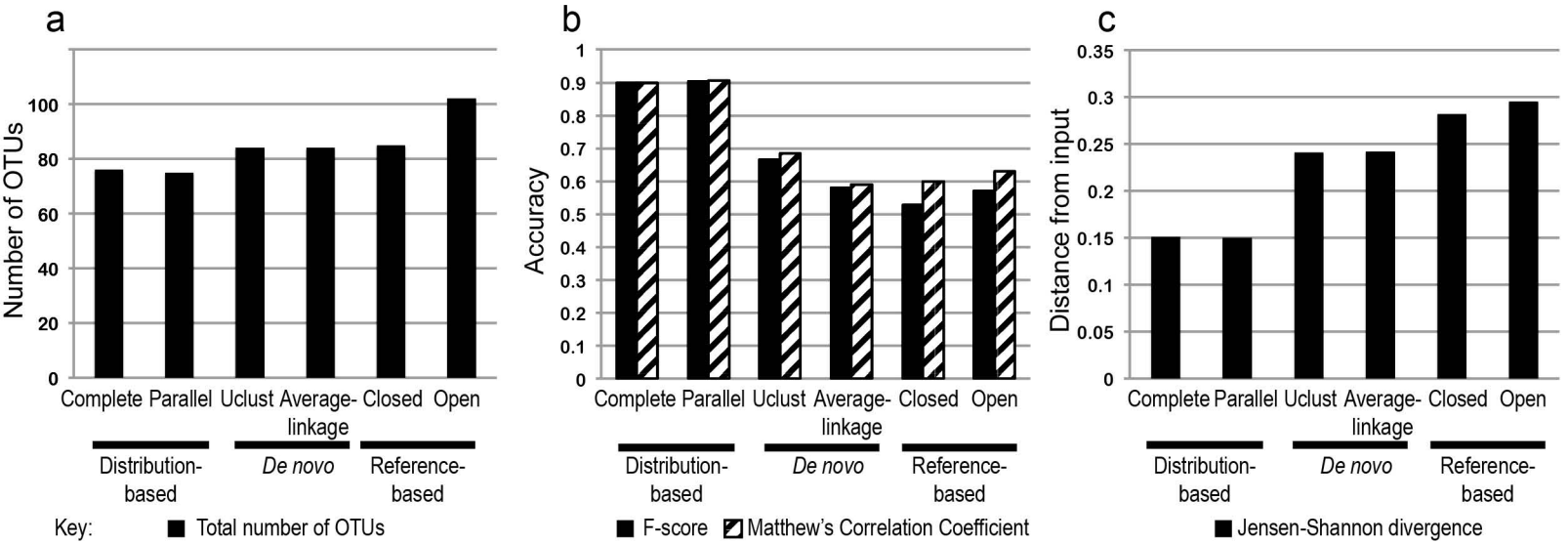
Lower Identity Threshold

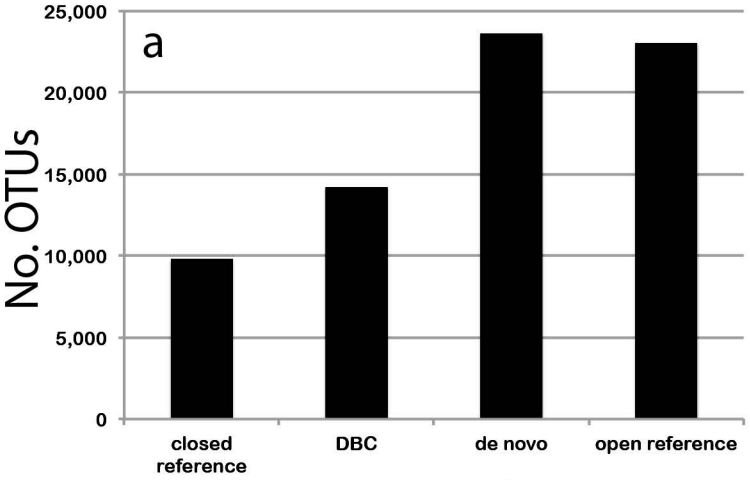
Desired Clustering

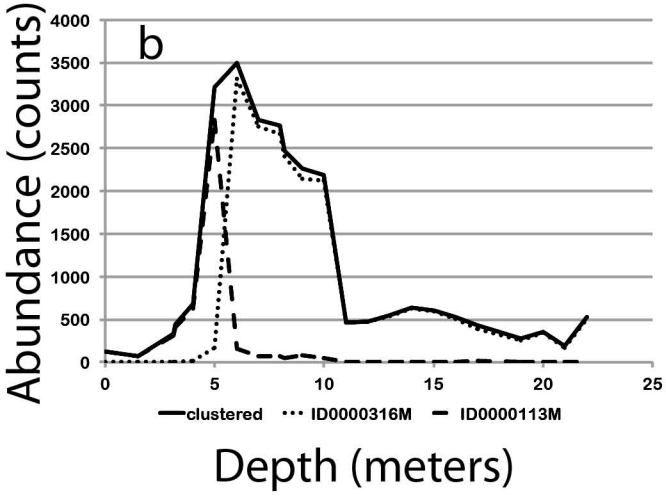



Pool of all sequences
Step 1
New candidate sequence (sort by decreasing abundance)
None
Step 2
New OTU from existing pool (sorted by decreasing distance of OTU rep. from candidate)
None
No
Step 3
Does OTU rep. satisfy **abudance criteria**?
Yes
Step 4
Is candidate **distributed differently** from OTU 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


a
Number of OTUs
40
35
30
25
20
15
10
5
0
700
600
500
400
300
200
100
0
Complete
Parallel
Uclust
Average-linkage
Closed
Open
Distribution-based
De novo
Reference-based
Key: Correct (left axis) Incorrect (right axis)
b
Accuracy
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Complete
Parallel
Uclust
Average-linkage
Closed
Open
Distribution-based
De novo
Reference-based
F-score
Matthew's Correlation Coefficient
c
Distance from input
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Complete
Parallel
Uclust
Average-linkage
Closed
Open
Distribution-based
De novo
Reference-based
Jensen-Shannon divergence


a
Number of OTUs
0
20
40
60
80
100
Complete
Parallel
Uclust
Average-linkage
Closed
Open
Distribution-based
De novo
Reference-based
Key:
Total number of OTUs
b
Accuracy
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Complete
Parallel
Uclust
Average-linkage
Closed
Open
Distribution-based
De novo
Reference-based
F-score
Matthew's Correlation Coefficient
c
Distance from input
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
Complete
Parallel
Uclust
Average-linkage
Closed
Open
Distribution-based
De novo
Reference-based
Jensen-Shannon divergence


a
No. OTUs
25,000
20,000
15,000
10,000
5,000
0
closed reference
DBC
de novo
open reference
Clustering Method


b
Abundance (counts)
4000
3500
3000
2500
2000
1500
1000
500
0
0
5
10
15
20
25
clustered
ID0000316M
ID0000113M
Depth (meters)

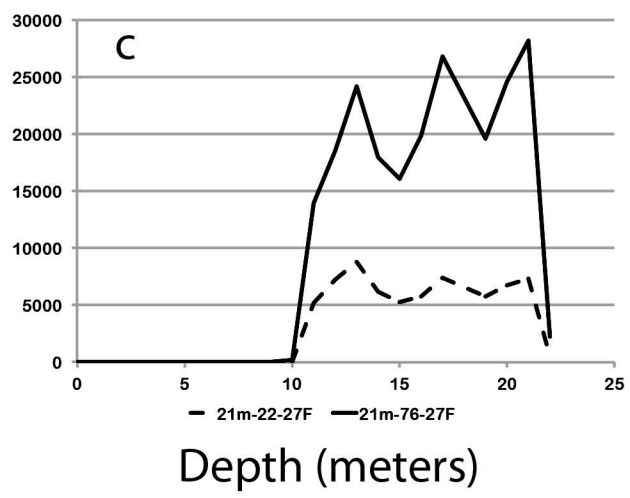

c
30000
25000
20000
15000
10000
5000
0
0
5
10
15
20
25
21m-22-27F
21m-76-27F
Depth (meters)

Table 1 The ability of each clustering algorithm to groups reads from the same input sequence together into the same OTU

	Distribution-based		*De novo*		Reference-based	
	Complete	Parallel	USEARCH	Average[5]	Open	Closed
TP[1]	8.57E+08	8.45E+08	9.46E+08	8.60E+08	6.61E+08	6.68E+08
FP[2]	6.86E+07	5.08E+07	7.36E+08	3.92E+08	1.32E+04	1.10E+04
TN[3]	1.48E+11	1.48E+11	1.48E+11	1.48E+11	1.48E+11	1.37E+11
FN[4]	2.02E+08	2.14E+08	1.13E+08	1.99E+08	3.98E+08	1.15E+10

[1] TP: True positives when two reads with best blast hits to the same mock community input sequence cluster in the same OTU

[2] FP: False positives when two reads with best blast hits to different mock community input sequences cluster in the same OTU

[3] TN: True negatives when two reads with best blast hits to different mock community input sequences cluster in different OTUs

[4] FN: False negative when two reads with best blast hits to the same mock community and in different OTUs or when one of the reads is not assigned to any OTUs (Closed-reference only)

[5] Average: Average-linkage hierarchical clustering

Table 2. Total number of OTUs remaining after filtering out low abundance OTUs

	Mock Community			Environmental Sample		
Method	*No filter*[2]	*>1*	*>10*	*No filter*	*>1*	*>10*
DBC[1] (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 (average - 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

[1] DBC: Distribution-based clustering

[2] 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

Table 3. Representative clustering times for mock community samples with various algorithms

	Total Run Time (hr:min:sec)[1]	
Clustering Method	*Mock community*[2]	*Environmental Sample*[3]
Distribution-based clustering (complete)	1:09:40	NA[4]
Distribution-based clustering (parallel)[5]	0:21:31	7:58:57
De novo (average 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

[1] Times are approximated by the difference between the start time and end time in the shell script examples in SI.
[2] The mock community contains 565,498 total reads and 5489 unique sequences
[3] The environmental sample contains 7,539,779 total reads and 120,601 unique sequences.
[4] NA indicates that this method was not performed.
[5] The distribution-based clustering algorithm was the only one was parallelized. 60-100 different processes were run at one time. Other methods would have had improved speeds if run in parallel.

Supplementary Information

Environmental sample collection

Water was collected from Upper Mystic Lake, (Winchester, MA, ~ 42 26.155N, 71 08. 961W) on Aug, 13, 2008 using a peristatic pump and plastic Tygon tubing. Tubing was lowered to a point ~1 m from the bottom, running the pump in reverse to prevent water from entering the tubing until the appropriate depth was reached. Water from depth was allowed to flow through the tubing for 5 minutes before 14 mls were collected into a 15 ml sterile falcon tube and immediately placed on dry ice. The first sample was taken from 22m depth and subsequent samples were taken every meter until 3m, then at 1.5m and the surface. Samples were transported on dry ice and stored at -80 °C until processing (about 1 year later).

DNA extraction

DNA was extracted as previously described (Blackburn 2010). Briefly, DNA was extracted with a modified version of the Qiagen DNeasy Blood and Tissue Kit (Qiagen, USA). Water was filtered through Swinnex filter holders onto 0.22 µm filters (Millipore, Billerica, MA). Filters were sterilely cut and added to a 2 ml screw cap tube containing 0.25 g of 0.1 mm zirconium/glass beads (MoBio Laboratories, Inc., Carlsbad, CA). 180 µl of lysis buffer consisting of 20 mM Tris HCl, 2 mM EDTA, and 1.2% Triton-X100 (pH 8.0) was added and samples were placed on a Mini Beadbeater-1 (BioSpec Products, Inc., Bartlesville, OK) for 1 minute at maximum speed. 180 µl of lysis buffer with 40 mg/ml lysozyme was added and the sample was incubated at 37 °C for 1 hr with shaking (450 rpm). 50 µl of proteinase K was added along with 400 µl of AL buffer (Qiagen DNeasy kit) without ethanol. Samples were mixed and incubated at 56 °C for 45 min. followed by a 5 min. incubation at 95 °C. Samples were centrifuged and the liquid was transferred to a sterile 1.5 ml tube. 400 µl of 100% ethanol was added and the liquid was added to the Qiagen DNeasy column. DNA was washed on the column following the Qiagen DNeasy protocol, including 500 µl wash with AW1 and AW2 and a final elution in 100 µl AE.

Mock community template preparation

To make the clone library, 16S rRNA sequences were amplified with Phusion polymerase (New England Biolabs, Ipswich, MA) and 27F and 1492R primers (Lane 1991). PCR products were cloned into the pCR Blunt II plasmid with the Zero Blunt TOPO PCR cloning kit (Invitrogen, Carlsbad, CA) and sequenced in at least one direction with Sanger sequencing (Genewiz, South Plainfield, NJ). Plasmids were purified using the plasmid DNA isolation reagent system (Carolina Biological Supply Co., Burlington, NC) and digested with restriction enzyme NotI (New England Biolabs, Ipswich, MA) to linearize the plasmid.

Illumina library preparation

Real-time PCR reactions were done first to normalize template concentrations and avoid cycling any templates past mid-log phase. PCR reactions for Illumina libraries were carried out as follows: 0.5 units of Phusion with 1 x High Fidelity buffer, 200 μM of each dNTP, 0.3 μM of PE16S_V4_U515_F and PE16S_V4_E786_R first step primers and approximately 40 ng of mixed DNA template were added for each 25 μl reaction. Additionally, 5 X SYBR Green I nucleic acid stain (Molecular Probes, Eugene, OR) was added for real-time PCR. Samples were cycled with the following conditions: denaturation at 98 °C for 30 sec annealing at 52 °C for 30 sec and extension at 72 °C for 30 sec. 14 cycles was mid-log for all samples and was subsequently used as the number of cycles for the first step PCR. The first step PCR reaction was cycled as four 25 μl reactions for each sample. PCR reactions were pooled and cleaned with Agencourt AMPure XP- PCR purification (Beckman Coulter, Brea, CA) according to the manufacture's protocol.

Illumina specific adaptors were added during a second step amplification. The conditions for the second step PCR were similar to the first step, although 4 μl of the purified first step reaction was used as a template and 0.4 μM of each PE-III-PCR-F and the barcoded reverse primer was used with 9 cycles. Samples were cycled as four 25 μl reactions and cleaned with Agencourt AMPure XP- PCR purification system. The nine libraries were sequenced in groups of three across three lanes (two flow cells) on both the Illumina GA II and HiSeq at the Biomicro

Center (MIT, Cambridge, MA) with 93 other samples per lane.

Calculation of error rate per sample

Raw data from reads with an exact match to one of the nine barcodes used for this experiment were used for comparing error rates across flow cells and lanes. This was necessary because these samples were multiplexed into lanes containing up to 93 additional unrelated samples. The raw, unfiltered fastq files were converted into a fasta file using a custom perl script. Blast was used to map the raw sequences to the mock community members, where the mock community database was trimmed to the amplified region between, but not including, the forward and reverse primer site. Raw sequences were only considered if the query and subject start and stop positions corresponded to the full length of the Illumina forward read. The perfect match, and single and double base mismatches, taken from the blast output, were calculated as a percent of the total that map to the full length Illumina sequence (Fig. S6). Sequences with less than 100% query or subject coverage were not considered in this calculation.

Commands used during processing

The following commands were used during processing.

Closed-reference clustering with QIIME (Shell):

```
#! /bin/sh
#$ -S /bin/bash
# -cwd

source /etc/profile.d/modules.sh
module load qiime-default
module load mothur
#fasta file name in QIIME format from first string after command
FASTAFILE=$1
#output directory as second string after command
OUTPUT=$2
```

```
#reference fasta file (latest greengenes OTUS)
REFERENCEFA~/greengenes/gg_12_10_otus/rep_set/97_otus.fasta
#reference taxonomies
REFERENCETAX=~/greengenes/gg_12_10_otus/taxonomy/97_otu_taxonomy.txt
PARAMS~/bin/methods_scripts/closed_ref_params.txt

echo "Start time"
date +"%m-%d-%y"
date +"%T"

pick_reference_otus_through_otu_table.py -o ${OUTPUT} -i ${FASTAFILE} -r
${REFERENCEFA} -t ${REFERENCETAX} -p ${PARAMS}

pick_rep_set.py --input ./${OUTPUT}/uclust_ref_picked_otus/*_otus.txt --
rep_set_picking_method most_abundant --fasta_file ${FAST
AFILE} -o ./${OUTPUT}/uclust_ref_picked_otus/otus_rep_set.fa

echo "End time"
date +"%m-%d-%y"
date +"%T"
```

Closed-reference QIIME parameters:

```
pick_otus:otu_picking_method uclust_ref
pick_otus:refseqs_fp  /greengenes/gg_12_10_otus/rep_set/97_otus.fasta
pick_otus:enable_rev_strand_match True
pick_otus:suppress_new_clusters      True
```

Open-reference clustering with QIIME (shell)

```
#! /bin/sh
#$ -S /bin/bash
```

```
# -cwd

source /etc/profile.d/modules.sh
module load qiime-default
module load mothur
#fasta file name in QIIME format
FASTAFILE=$1
#output folder (unique)
OUTPUT=$2
#reference fasta file (latest greengenes OTUS)
REFERENCEFA=/data/spacocha/Qiime_dir/greengenes/gg_12_10_otus/rep_set/97
_otus.fasta
#reference taxonomies
REFERENCETAX=/data/spacocha/Qiime_dir/greengenes/gg_12_10_otus/taxonomy
/97_otu_taxonomy.txt
PARAMS=/home/spacocha/bin/methods_scripts/open_ref_params.txt

echo "Start time"
date +"%m-%d-%y"
date +"%T"


pick_reference_otus_through_otu_table.py -o ${OUTPUT} -i ${FASTAFILE} -r
${REFERENCEFA} -t ${REFERENCETAX} -p ${PARAMS}

pick_rep_set.py --input ./${OUTPUT}/uclust_ref_picked_otus/*_otus.txt --
rep_set_picking_method most_abundant --fasta_file ${FAST
AFILE} -o ./${OUTPUT}/uclust_ref_picked_otus/otus_rep_set.fa

echo "End time"
date +"%m-%d-%y"
```

```
date +"%T"
```

Open-reference QIIME parameters:

```
pick_otus:otu_picking_method uclust_ref
pick_otus:refseqs_fp greengenes/gg_12_10_otus/rep_set/97_otus.fasta
pick_otus:enable_rev_strand_match True
pick_otus:suppress_new_clusters      False
```

De novo USEARCH (shell)

```
#! /bin/sh
#$ -S /bin/bash
# -cwd

#fastafile
FASTAFILE=$1
#matfile
MATFILE=$2

echo "Start time"
date +"%m-%d-%y"
date +"%T"

perl ~/bin/fasta2uchime_mat.pl ${MATFILE} ${FASTAFILE} > ${FASTAFILE}.ab
~/bin/usearch6.0.307_i86linux32 -cluster_fast ${FASTAFILE}.ab -id 0.97 -uc
${FASTAFILE}.uc
perl ~/bin/UC2list2.pl ${FASTAFILE}.uc > ${FASTAFILE}.list
perl ~/bin/list2mat.pl ${MATFILE} ${FASTAFILE}.list eco > ${FASTAFILE}.list.mat
perl ~/bin/fasta2filter_from_mat.pl  ${UNIQUE}.list.mat ${FASTAFILE} >
${FASTAFILE}.list.mat.fa
```

```
echo "End time"
date +"%m-%d-%y"
date +"%T"
```

Mothur command (batch)

```
unique.seqs(fasta=unique.uchime.remove.tocluster.fa)
align.seqs(fasta=unique.uchime.remove.tocluster.unique.fa,
reference=/data/spacocha/tmp/silva.bacteria.fasta)
screen.seqs(fasta=unique.uchime.remove.tocluster.unique.align,
name=unique.uchime.remove.tocluster.names, start=13862,
end=15958,minlength=76)
filter.seqs(fasta=unique.uchime.remove.tocluster.unique.good.align, vertical=T,
trump=.)
unique.seqs(fasta=unique.uchime.remove.tocluster.unique.good.filter.fasta,
name=unique.uchime.remove.tocluster.good.names)
system(cp unique.uchime.remove.tocluster.unique.good.filter.unique.names
final.names)
system(cp unique.uchime.remove.tocluster.unique.good.filter.names final.names)
dist.seqs(fasta=final.fasta, cutoff=0.15)
cluster(column=final.dist, name=final.names)
```

Generation of principal component analysis plots

Principal component analysis was done on the final OTU by library matrices for each clustering algorithm using QIIME beta_diversity_through_plots.py. The lowest number of sequences in a library was determined using QIIME's per_library_stats.py and input into beta_diversity_through_plots.py (-e). Trees of the representative samples were made with FastTree.

Simulated mock community data with varying error rates across libraries

To determine the impact of different error rates across libraries on distribution-based clustering performance, simulated mock community was generated using the template sequences for each members added across libraries. The total number of sequences generated was proportional to measured concentration and resulted in the creation of 748,463 total *in silico* reads. The geometric mean (R version 2.12.1; rgeom) was used to create error rates of both 0.9 and 0.8 to simulate high and low quality sequencing runs, respectively. The constant error rate dataset used in Table S5 was 0.9 for all libraries while the variable error rate dataset was 0.90 for 6 libraries and 0.08 for 3 of the libraries.

The geometric mean was used to determine which of the simulated reads would contain errors and how many errors it would contain. This was implemented in R (version 2.12.1) with rgeom using the total read count needed for each sequence and the error rate. For example, if a template was supposed to have 10 reads with an error rate of 0.8, the results would look similar to the following:

```
> rgeom(10,0.8)
 [1] 1 0 0 0 0 0 1 4 0 0
```

Where two sequences would have one bp different, one would have four mismatches and seven sequences would have no errors.

After determining how many errors to generate for each read, the position of the errors was also determined in R using the hypergeometric mean (rhyper). The distribution results in either 0 or 1 and depends on the input probability. Starting at the 3' ending position, the hypergenometric mean was used to determine whether to alter the base to another random base (1=alter, 0=evaluate next base). The probability of having an error decreased toward the 5' end to mimic sequence quality being poor at the 3' end. This was repeated until the required number of errors was generated.

Two datasets were generated in this manner. One set had a constant error rate across all libraries, and another had three libraries with a higher error rate. The dataset was clustered using the distribution based clustering algorithm as normal and the results are presented in Table S5.

Supplementary References

Blackburn MC (2010). Development of New Tools and Applications for High-Throughput Sequencing of Microbiomes in Environmental or Clinical Samples. Master of Science in Chemical Engineering thesis, Massachusetts Institute of Technology, Cambridge, MA.

Lane DJ (1991). 16S/23S rRNA sequencing. In: Stackebrandt E, Goodfellow M (eds). *Nucleic Acid Techniuqes in Bacterial Systemantics*. Wiley & Sons: Chinchester. pp 115-175.

Supplementary Figures

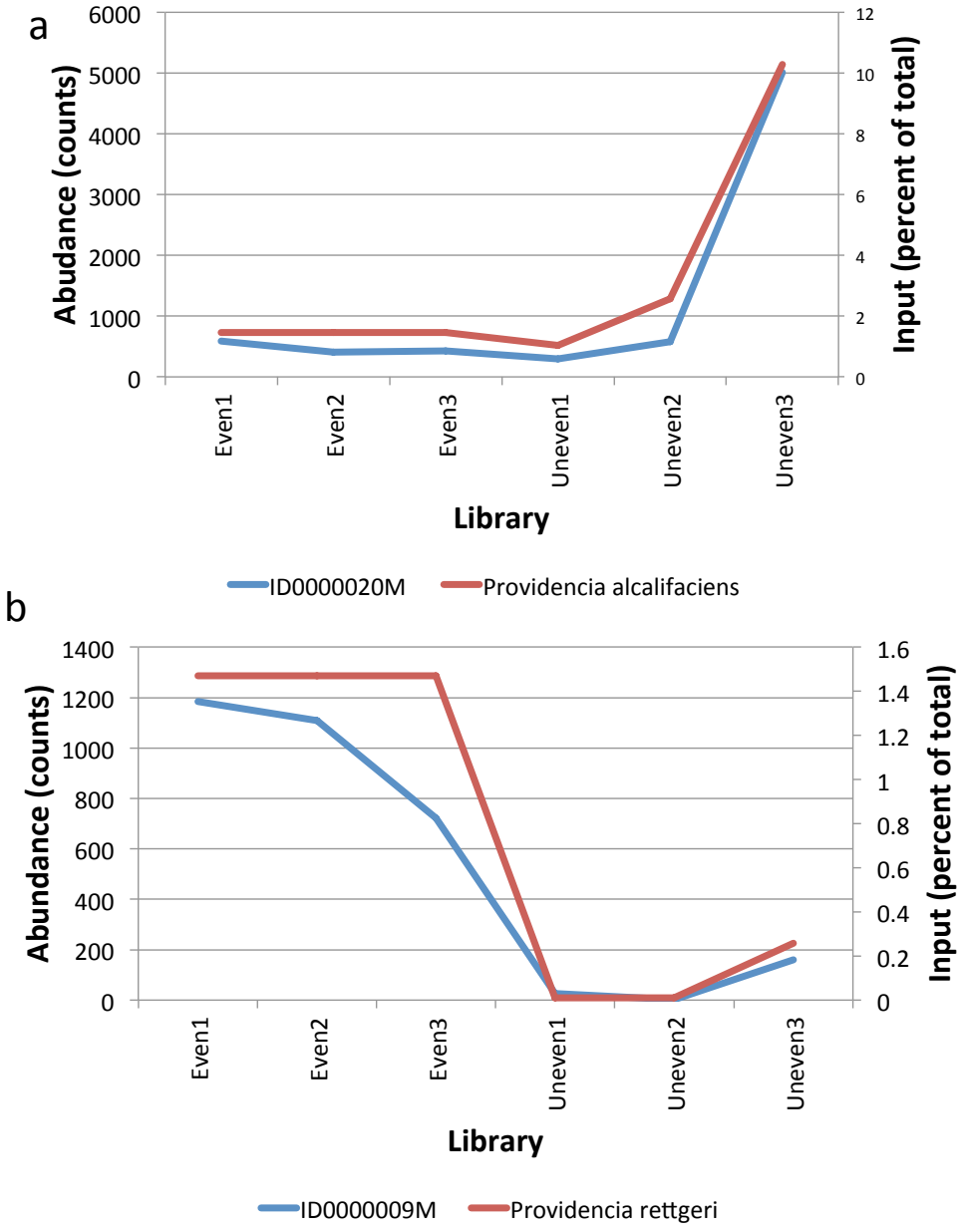


Figure S1. Unique distributions across the mock libraries help to identify a mislabeled sequence in the reference dataset. a. The distribution of a sequence matching the reference sequence labeled *Providencia rettgeri* and the input distribution of *Providencia alcalifaciens*. This sequence also matched others strains labeled *Providencia alcalifaciens* in NCBI's nr database*.* It was changed to *Providencia alcalifaciens.* b.) The distribution of another sequence which corresponds to the correct input of *Providencia rettgeri.* This other sequence also hits many other *Providencia rettgeri* strain in NCBI's nr database. This sequence was included in the analysis as the reference sequence for *Providencia rettgeri.*

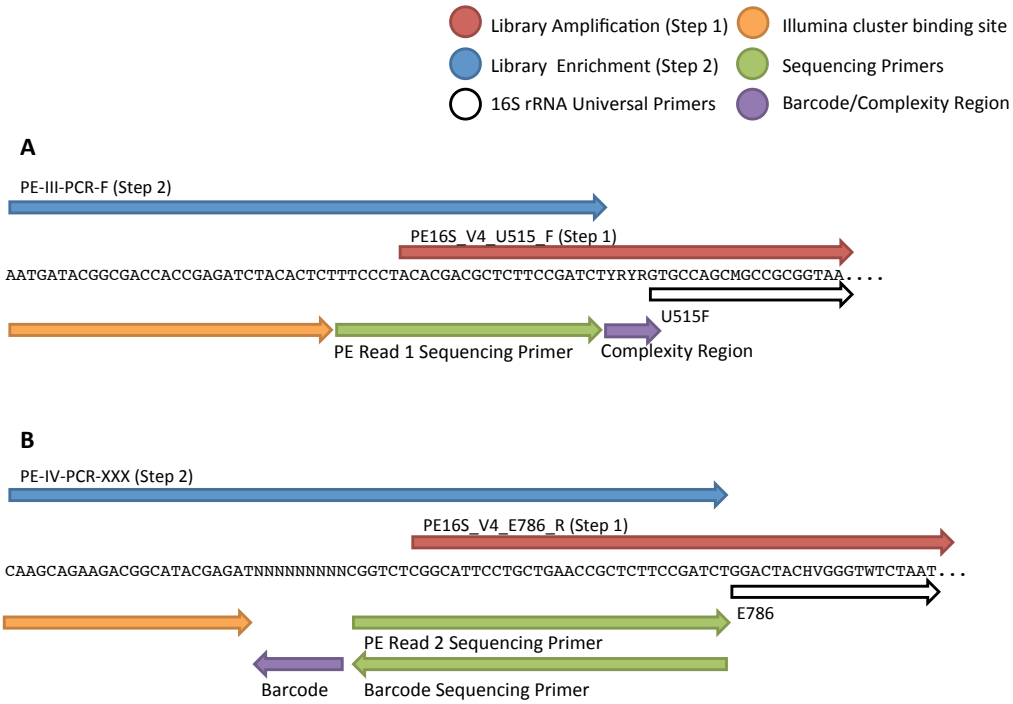


Figure S2. 16S rDNA library construct from two-step PCR. a.) 5' end of the Illumina library construct, including both first and second step forward primer sequences and sequencing primers. b.) 3' end of Illumina library construct including barcoded region and first step and second step reverse primers.

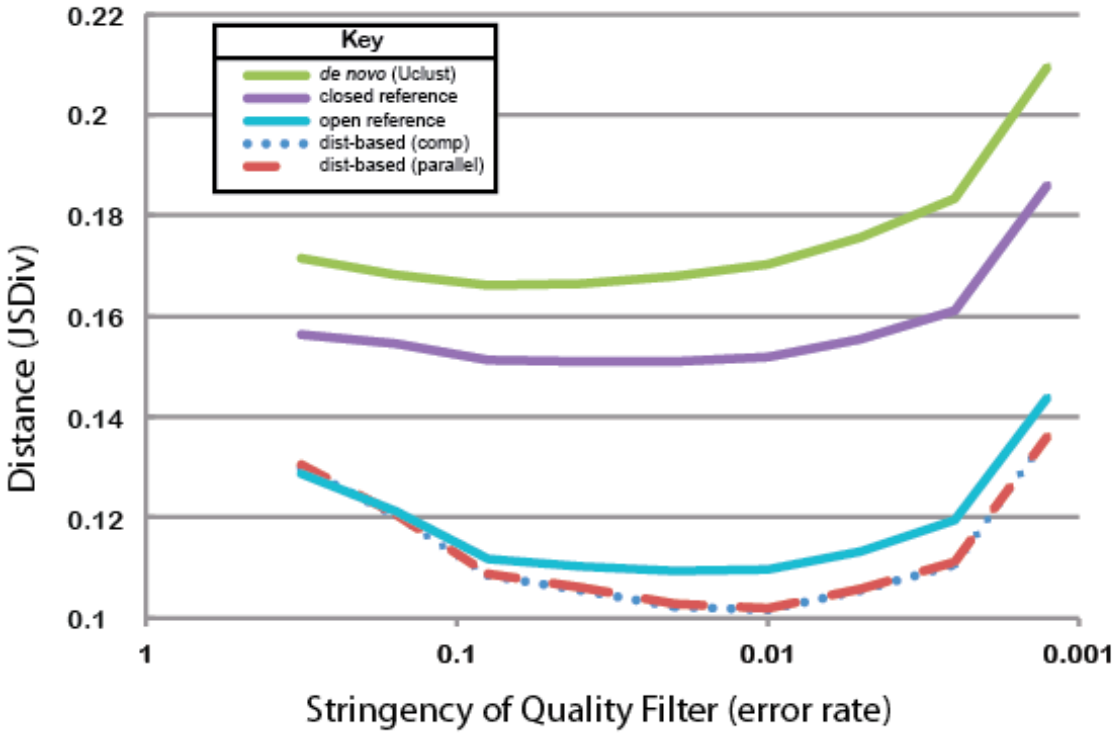


Figure S3. Distribution-based clustering results in a distribution of sequences that is most similar to the input distribution. Additionally, an intermediate amount of quality filtering results in a better representation of the input community for all clustering methods. The Jensen-Shannon divergence (JSDiv) is used as a measure of distance between the input concentration and resulting OTU counts after applying each clustering method at different levels of quality filtering. At the highest error rates, incorrect OTUs add to the distance from the true distribution. At the lowest error rates, the small number of reads kept creates the large distance values. Both parallel and complete distribution-based clustering methods result in OTUs that are most similar to the true distribution at intermediate levels of quality filter stringency.

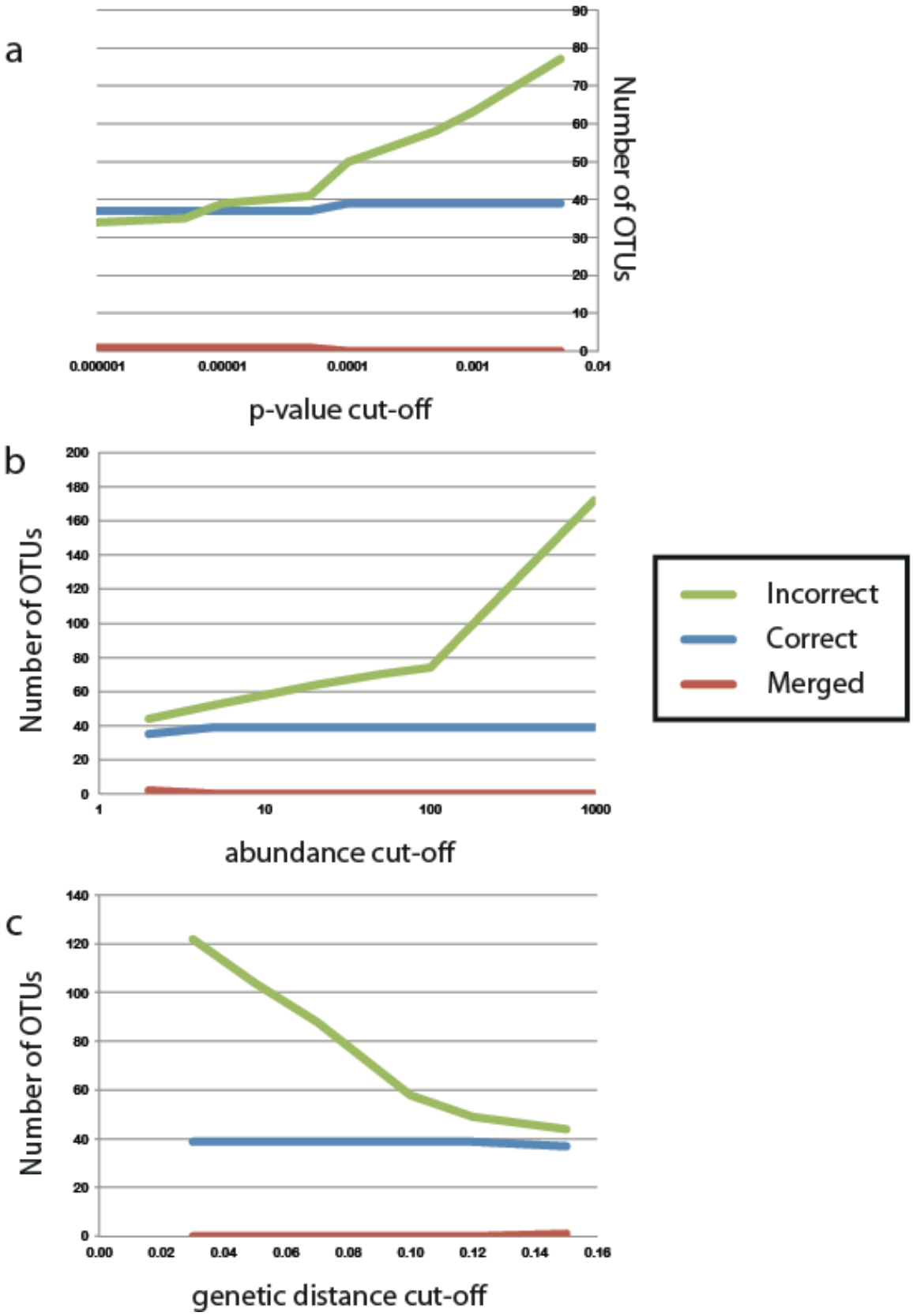
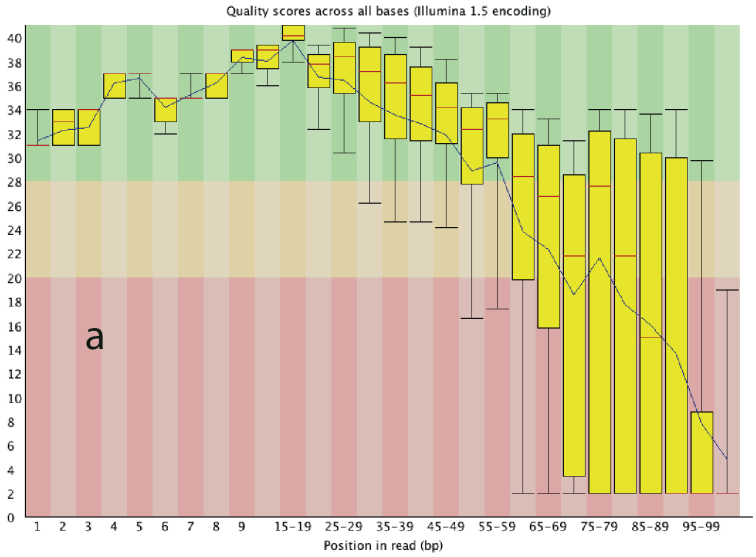
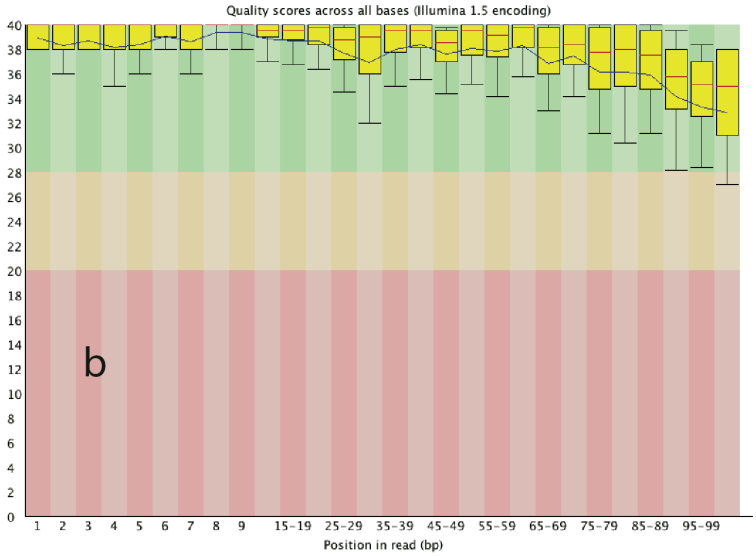

a
90
80
70
60
50
40
30
20
10
0
Number of OTUs
0.000001
0.00001
0.0001
0.001
0.01
p-value cut-off
b
200
180
160
140
120
100
80
60
40
20
0
Number of OTUs
1
10
100
1000
abundance cut-off
Incorrect
Correct
Merged
c
140
120
100
80
60
40
20
0
Number of OTUs
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
genetic distance cut-off

Figure S4. Sensitivity of the resulting OTUs to changes in the distribution-based algorithm parameters. a.) Increasing the significance cut-off value of the chi-sq test creates more incorrect OTUs whereas lower p-value cut-offs tend to merge sequences. X- axis in plotted in log scale b.) Decreasing the abundance criteria merges true input sequences with similar distributions, but increasing the cut-off to 10 mainly detects sequencing errors. X-axis is plotted in log scale c.) Lower genetic similarity cutoffs generate more incorrect OTUs, whereas at high genetic cut-off values, some mock community sequences with similar distributions are merged. "Correct" are the number of OTUs containing a single exact match to an input sequence. "Incorrect" are the number of OTUs that do not have any sequences exactly matching the input community. "Merged" are the number of OTUs that contain more than one sequence matching an input sequence.


Quality scores across all bases (Illumina 1.5 encoding)
a
Position in read (bp)


Quality scores across all bases (Illumina 1.5 encoding)
b
Position in read (bp)

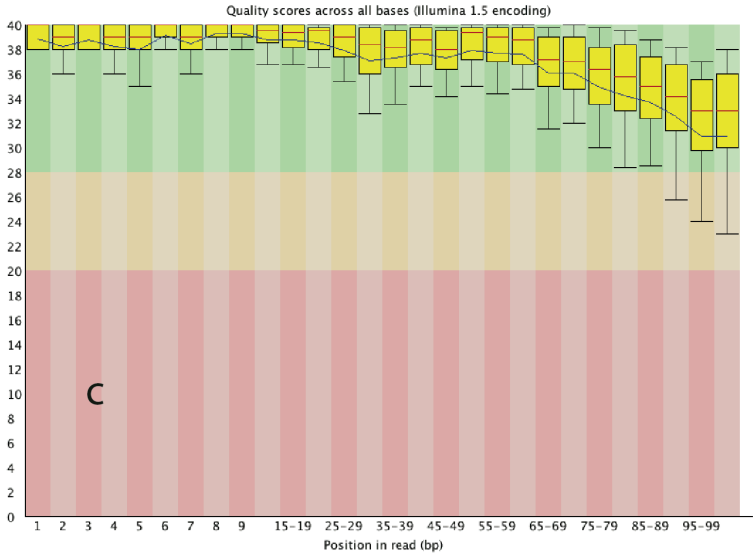

Quality scores across all bases (Illumina 1.5 encoding)
c
Position in read (bp)

Figure S5. Per base quality scores for the three lanes of Illumina. The quality of one set of samples was substantially worse than the others. (a) Flow 1, Lane 1, samples com4-com6 (b) Flow 2, Lane 1, samples com1-com3 (c) Flow 2, Lane 2, samples com7-com9.

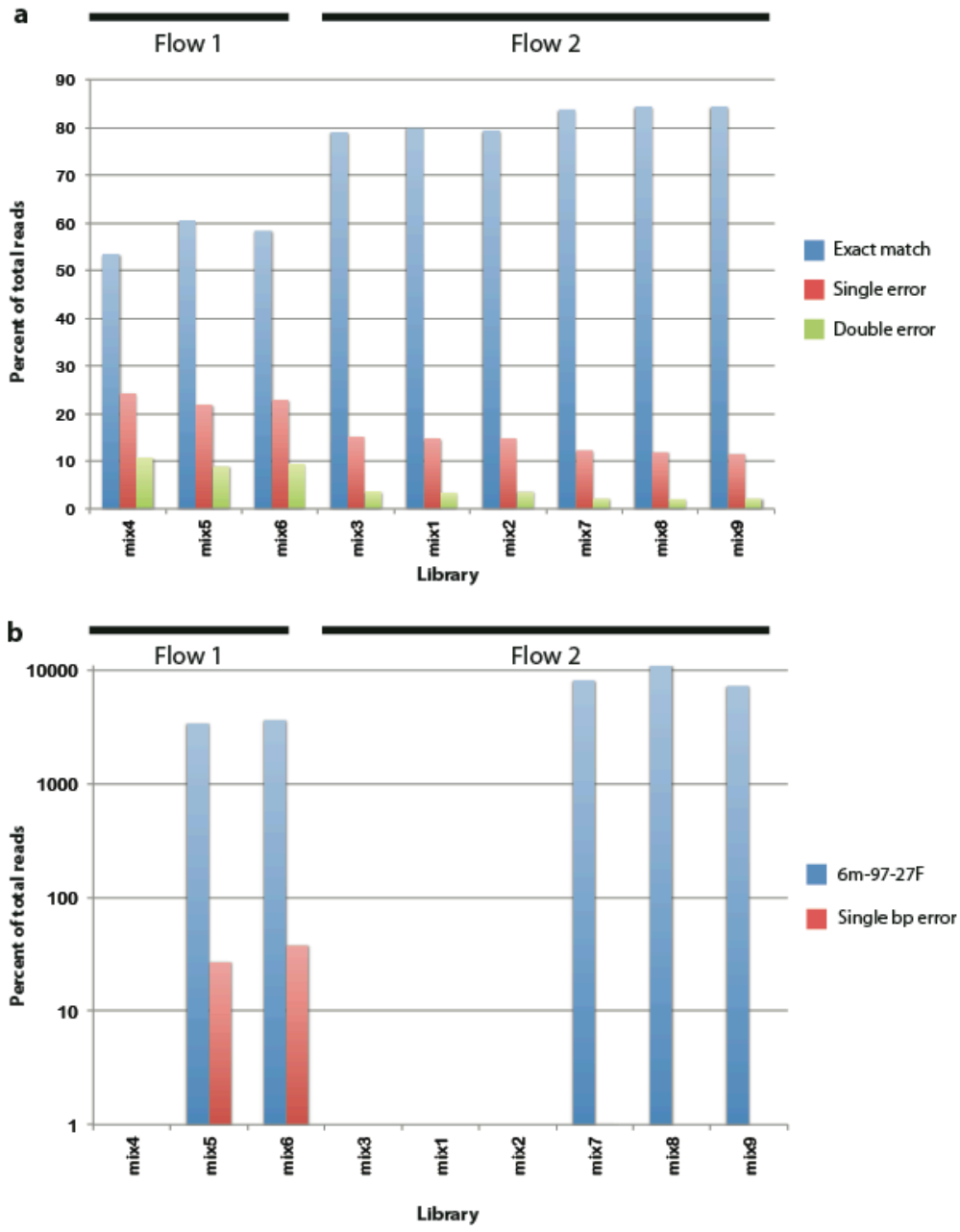
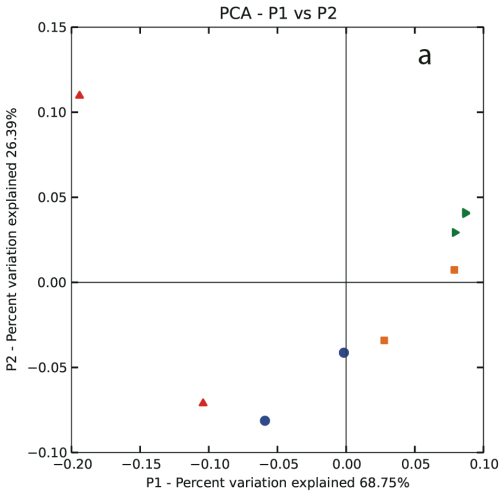
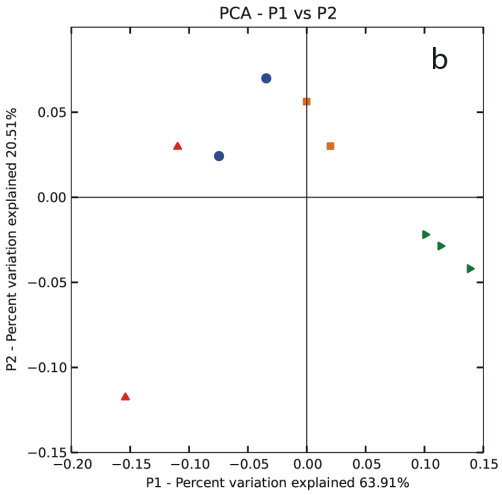
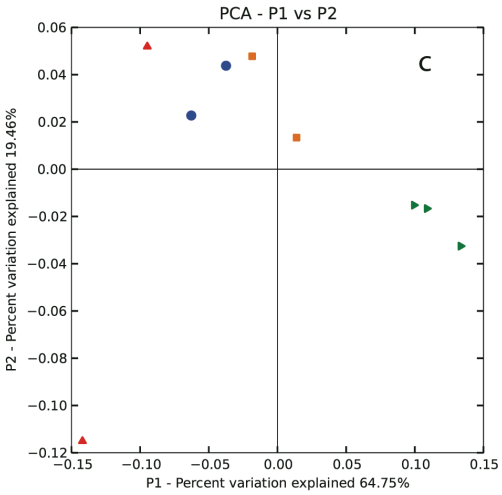


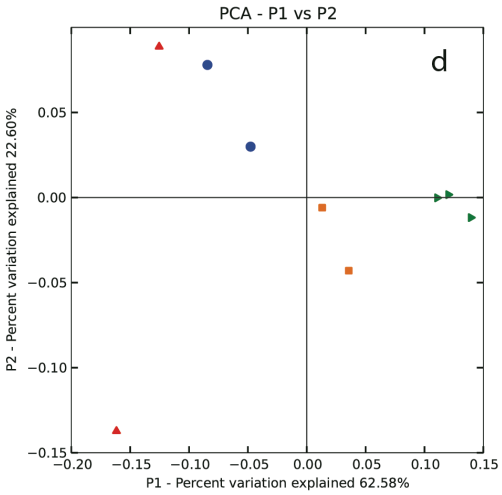
Figure S6. Error rates were higher on Flow cell 1 (Flow1) than Flow cell 2 (Flow2) causing non-random distribution of erroneous sequences across samples. (a) Exact match, single base mismatches (single error) and double base mismatches (double

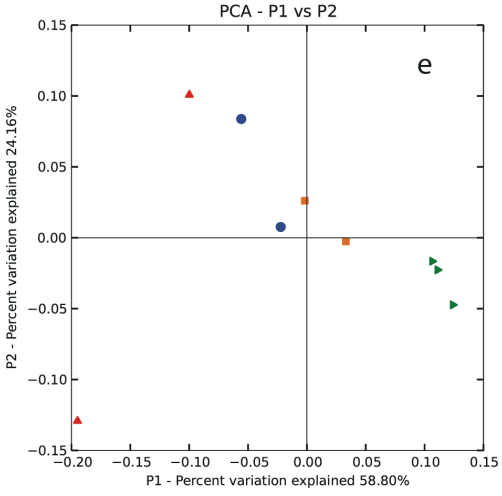
error) as a percent of the total number of raw (not quality filtered) sequences that blast to the entire 76 bp of any mock community member. Samples are labeled with the flow cell number (Flow1 or Flow2) corresponding to two different Illumina runs. Additionally, the corresponding sample name (mix1-9) is labeled on the X-axis. (b) The distribution of the true sequence (6m-94-27F) and a sequence with a single bp error sequence across samples after quality filtering and clustering. Although the single bp error sequence was generated from the true sequence, it does not have the same distribution across samples because of the difference in error rates across flow cells. Y-axis is log scale.


PCA - P1 vs P2
a
P2 - Percent variation explained 26.39%
P1 - Percent variation explained 68.75%


PCA - P1 vs P2
b
P2 - Percent variation explained 20.51%
P1 - Percent variation explained 63.91%


PCA - P1 vs P2
c
P2 - Percent variation explained 19.46%
P1 - Percent variation explained 64.75%


PCA - P1 vs P2
d
P2 - Percent variation explained 22.60%
P1 - Percent variation explained 62.58%


PCA - P1 vs P2
e
P2 - Percent variation explained 24.16%
P1 - Percent variation explained 58.80%

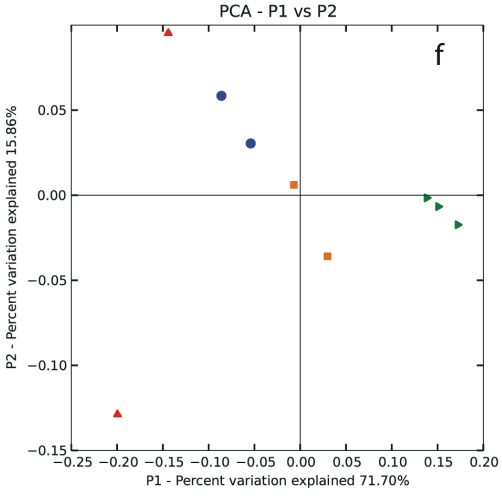

PCA - P1 vs P2
f
P2 - Percent variation explained 15.86%
P1 - Percent variation explained 71.70%

Figure S7. Principal components analysis of mock community libraries com1-com9. The primary (P1) and secondary (P2) components are plotted for the (a) true input community and for each clustering method: (b) distribution-based clustering, complete; (c) distribution-based clustering, parallel; (d) *de novo,* usearch; (e) open-reference clustering; (f) closed-reference clustering. Samples are colored according to the total number of input sequences: 1-10 input sequences, red triangle; 11-20, blue circle; 21-30 orange square; 31-40 green triangle.

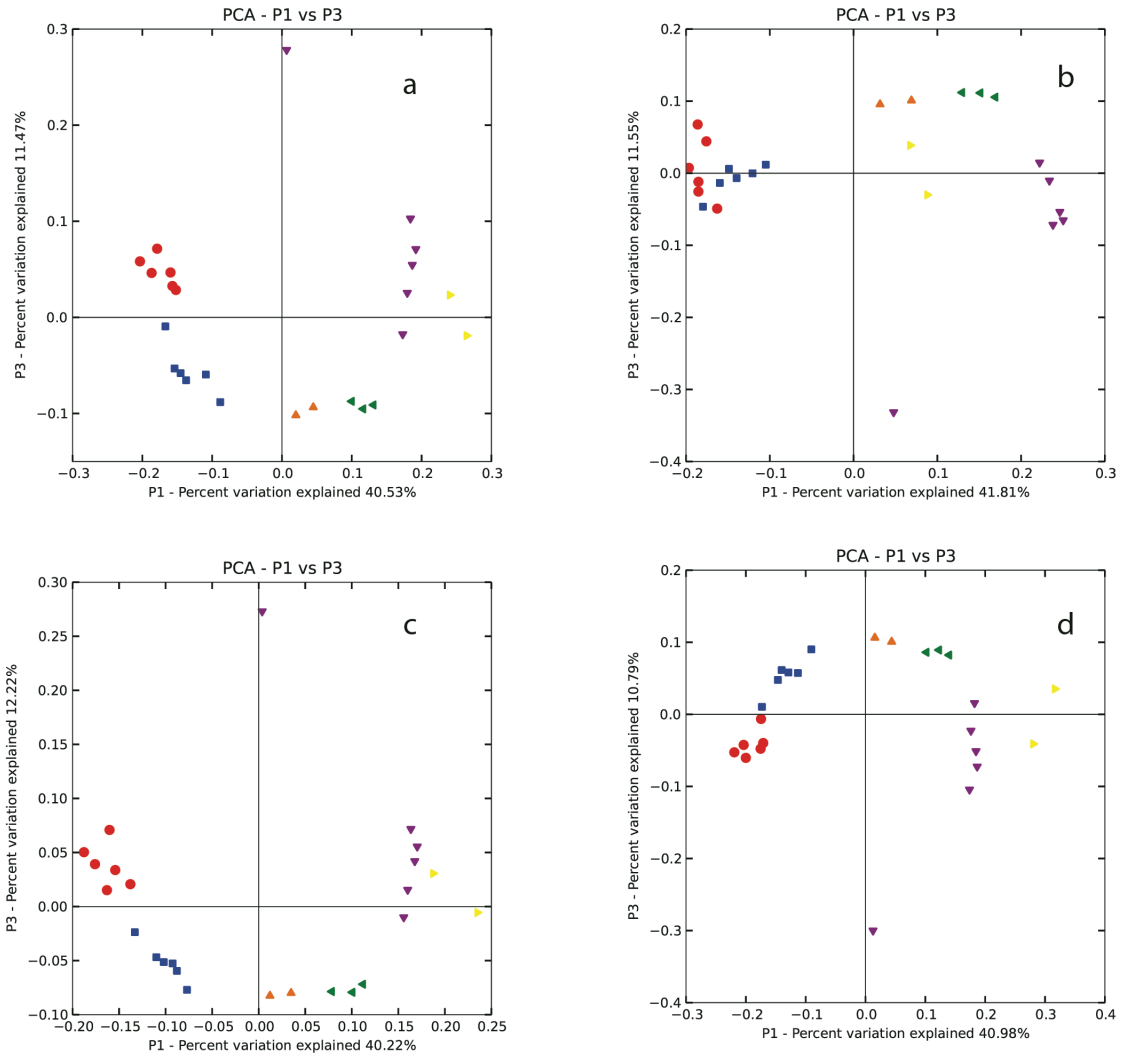


Figure S8. Principal components analysis of environmental samples from a stratified lake is similar across clustering algorithms. The primary (P1) and tertiary (P3) components are plotted for (a) distribution-based clustering, (b) *de novo* (USEARCH), (c) closed-reference and (d) open-reference clustering. Samples are colored according to depth: surface to 5 meters (m) depth, red circles; 6-10 m, blue square; 11-12 orange triangle; 13-15 m, green triangle; 16-22 m, purple triangle; surface and end blank, yellow triangle.

Supplementary Tables

Table S1. Mock community template concentrations and primer mismatches

Name	Set no.	Added to libs.[1]	Concentration (pg/ul)	Notes
21m-94-27F	1	com1-com9	22.67	NA
6m-05-27F	1	com1-com9	4.9	NA
6m-16-27F	1	com1-com9	1.988	NA
6m-10-27F	1	com1-com9	26.36	NA
21m-66-27F	1	com1-com9	38.07	2 mismatches 8 and 9 bp from 3' end of F primer
21m-32-27F	2	com2-com9	60.4875	NA
6m-09-27F	2	com2-com9	7.5625	NA
6m-44-27F	2	com2-com9	6.95	1 mismatch 1 bp from 3' end of F primer
6m-06-27F	2	com2-com9	17.6875	NA
21m-41-27F	2	com2-com9	39.5625	An additional 21.575 pg/ul of 21m-41 was added to com7, com8 and com9 as a mislabeled template.
6m-80-27F	3	com3-com9	8.0735	1 mismatch 13 bp from 3' end of the R primer
21m-90-27F	3	com3-com9	1.77	1 bp mismatch 9 bps from 3' end of F primer
6m-70-27F	3	com3-com9	1.99875	NA
6m-89-27F	3	com3-com9	28.85	NA
21m-02-27F	4	com4-com9	1.6525	2 bp mismatch 8 and 9 bp from 3' end of F primer
6m-22-27F	4	com4-com9	19	NA
6m-69-27F	4	com4-com9	8.625	NA
6m-50-27F	4	com4-com9	47.825	NA
21m-83-27F	4	com4-com9	56.05	1 mismatch 14 bp from 3' end R primer
6m-94-27F	5	com5-com9	59.775	NA
21m-25-27F	5	com5-com9	19.5	NA
21m-29-27F	5	com5-com9	38.7875	NA
21m-05-27F	5	com5-com9	10.3375	NA
6m-86-27F	6	com6-com9	0.60475	NA
21m-87-27F	6	com6-com9	36.2	1 mismatch 7 bp from 3' end of R primer
6m-65-27F	6	com6-com9	40.3625	1 mismatch 1 bp from 3' end F primer

21m-61-27F	6	com6-com9	94.7125	4 bps mismatch 12, 8, 7 and 2 bp from 3' end of F primer
6m-04-27F	7	com7-com9	3.657	NA
21m-54-27F	7	com7-com9	30	NA
6m-20-27F	7	com7-com9	17.175	NA
6m-40-27F	7	com7-com9	26.8625	NA
21m-08-27F	8	com8-com9	0.5205	2 mismatch 8 and 9 bp from 3' end of F primer
6m-81-27F	8	com8-com9	40.2875	NA
6m-13-27F	8	com8-com9	26.1625	NA
6m-52-27F	8	com8-com9	1.2065	NA
6m-75-27F	9	com9	16.675	NA
6m-82-27F	9	com9	22.675	NA
21m-68-27F	9	com9	13.875	NA
6m-19-27F	9	com9	8.85	NA
6m-87-27F	9	com9	0.8125	NA

[1] Samples were added to libraries in sequential order, starting with com1 and ending with com9. If a set was added to com1, it was also added to all subsequent libraries com2 through com9.

Table S2. Expected relative amounts of each DNA template per library

Name	Conc. (pg/ul)	com1	com2	com3	com4	com5	com6	com7	com8	com9
21m-94-27F	22.67	2.41E-01	1.11E-01	9.61E-02	6.62E-02	5.09E-02	4.78E-02	3.52E-02	3.24E-02	3.02E-02
6m-05-27F	4.9	5.21E-02	2.41E-02	2.08E-02	1.43E-02	1.10E-02	1.03E-02	7.60E-03	7.01E-03	6.54E-03
6m-16-27F	1.988	2.12E-02	9.77E-03	8.43E-03	5.81E-03	4.47E-03	4.19E-03	3.08E-03	2.84E-03	2.65E-03
6m-10-27F	26.36	2.80E-01	1.30E-01	1.12E-01	7.70E-02	5.92E-02	5.55E-02	4.09E-02	3.77E-02	3.52E-02
21m-66-27F	38.07	4.05E-01	1.87E-01	1.61E-01	1.11E-01	8.55E-02	8.02E-02	5.90E-02	5.44E-02	5.08E-02
21m-32-27F	48.39	0.00E+00	2.38E-01	2.05E-01	1.41E-01	1.09E-01	1.02E-01	7.50E-02	6.92E-02	6.45E-02
6m-09-27F	6.05	0.00E+00	2.97E-02	2.56E-02	1.77E-02	1.36E-02	1.27E-02	9.38E-03	8.65E-03	8.07E-03
6m-44-27F	5.56	0.00E+00	2.73E-02	2.36E-02	1.62E-02	1.25E-02	1.17E-02	8.62E-03	7.95E-03	7.42E-03
6m-06-27F	14.15	0.00E+00	6.96E-02	6.00E-02	4.13E-02	3.18E-02	2.98E-02	2.19E-02	2.02E-02	1.89E-02
21m-41-27F	35.2475	0.00E+00	1.73E-01	1.49E-01	1.03E-01	7.92E-02	7.43E-02	5.47E-02	5.04E-02	4.70E-02
6m-80-27F	6.4588	0.00E+00	0.00E+00	2.74E-02	1.89E-02	1.45E-02	1.36E-02	1.00E-02	9.24E-03	8.62E-03
21m-90-27F	1.416	0.00E+00	0.00E+00	6.00E-03	4.13E-03	3.18E-03	2.98E-03	2.20E-03	2.02E-03	1.89E-03
6m-70-27F	1.599	0.00E+00	0.00E+00	6.78E-03	4.67E-03	3.59E-03	3.37E-03	2.48E-03	2.29E-03	2.13E-03
6m-89-27F	23.08	0.00E+00	0.00E+00	9.78E-02	6.74E-02	5.18E-02	4.86E-02	3.58E-02	3.30E-02	3.08E-02
21m-02-27F	1.322	0.00E+00	0.00E+00	0.00E+00	3.86E-03	2.97E-03	2.79E-03	2.05E-03	1.89E-03	1.76E-03
6m-22-27F	15.2	0.00E+00	0.00E+00	0.00E+00	4.44E-02	3.41E-02	3.20E-02	2.36E-02	2.17E-02	2.03E-02
6m-69-27F	6.9	0.00E+00	0.00E+00	0.00E+00	2.01E-02	1.55E-02	1.45E-02	1.07E-02	9.87E-03	9.20E-03
6m-50-27F	38.26	0.00E+00	0.00E+00	0.00E+00	1.12E-01	8.59E-02	8.06E-02	5.93E-02	5.47E-02	5.10E-02
21m-83-27F	44.84	0.00E+00	0.00E+00	0.00E+00	1.31E-01	1.01E-01	9.45E-02	6.95E-02	6.41E-02	5.98E-02
6m-94-27F	47.82	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.07E-01	1.01E-01	7.42E-02	6.84E-02	6.38E-02
21m-25-27F	15.6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.50E-02	3.29E-02	2.42E-02	2.23E-02	2.08E-02
21m-29-27F	31.03	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.97E-02	6.54E-02	4.81E-02	4.44E-02	4.14E-02
21m-05-27F	8.27	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.86E-02	1.74E-02	1.28E-02	1.18E-02	1.10E-02
6m-86-27F	0.4838	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.02E-03	7.50E-04	6.92E-04	6.45E-04
21m-87-27F	28.96	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.10E-02	4.49E-02	4.14E-02	3.86E-02
6m-65-27F	32.29	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.01E-02	4.62E-02	4.31E-02
21m-61-27F	75.77	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.18E-01	1.08E-01	1.01E-01
6m-04-27F	2.9256	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.54E-03	4.18E-03	3.90E-03
21m-54-27F	24	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.72E-02	3.43E-02	3.20E-02
6m-20-27F	13.74	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.13E-02	1.96E-02	1.83E-02
6m-40-27F	21.49	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.33E-02	3.07E-02	2.87E-02
21m-08-27F	0.4164	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	5.95E-04	5.55E-04
6m-81-27F	32.23	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	4.61E-02	4.30E-02
6m-13-27F	20.93	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.99E-02	2.79E-02
6m-52-27F	0.9652	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.38E-03	1.29E-03
6m-75-27F	13.34	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.78E-02
6m-82-27F	18.14	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.42E-02
21m-68-27F	11.1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.48E-02
6m-19-27F	7.08	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.44E-03
6m-87-27F	0.65	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	8.67E-04

Table S3. Barcode sequences and sequencing outline

Sample ID	Barcode Sequence	Description	Flow cell No.	Lane No.	Diversity
com1	CGAATAT	E8, plate 63umP2	1	1	
com2	AAGGAAC	E9, plate 63umP2	1	1	
com3	GATTGAA	E10, plate 63umP2	1	1	
com4	CCGCACC	H1, plate 63umP1	2	1	
com5	ATGCCAG	H2, plate 63umP1	2	1	
com6	TCGAACA	H3, plate 63umP1	2	1	
com7	GTACGTT	H10, plate 63umP3	1	2	
com8	AGTAGAT	H11, plate 63umP3	1	2	
com9	TCATTAA	H12, plate 63umP3	1	2	

Table S4. Correlation of OTUs from various clustering methods with matching Sanger environmental clone sequence

Sanger Clone	USEARCH correlation	Open-reference correlation	Closed-reference correlation	DBC correlation
21m-02-27F	0.999831867	0.988437304	NA	NA
21m-03-27F	0.999999465	0.999768514	0.999768514	0.999953024
21m-04-27F	0.982567028	0.999991003	0.999991003	0.999968253
21m-05-27F	0.99998637	0.999952053	NA	0.999967953
21m-08-27F	0.998120162	0.996556495	NA	0.999521037
21m-09-27F	0.999922215	0.999952171	0.999952171	0.999858585
21m-11-27F	0.997205609	0.994006215	NA	0.947095652
21m-13-27F	0.999971592	0.999971592	0.999971592	0.999973636
21m-14-27F	NA	NA	NA	NA
21m-22-27F	0.999999929	0.999995556	0.999995763	0.999993992
21m-24-27F	0.999936853	0.999945906	NA	0.999948108
21m-29-27F	0.999999379	0.999996912	0.999996912	0.999995608
21m-30-27F	0.999044469	0.999185914	NA	0.99998369
21m-31-27F	0.999999836	0.999997084	0.999997084	0.999997146
21m-32-27F	0.999995376	0.999458687	0.999458687	0.999969546
21m-36-27F	0.999999722	0.999964949	NA	0.999955969
21m-40-27F	0.999999988	0.999998671	0.999998649	0.99999804
21m-41-27F	NA	0.997646314	0.997646314	0.999758171
21m-45-27F	0.999999156	NA	NA	0.99995629
21m-48-27F	0.976110722	1	0.99750752	0.976621605
21m-49-27F	0.999982722	NA	NA	0.999980964
21m-52-27F	NA	NA	NA	0.999698894
21m-60-27F	0.999999952	0.998506324	0.998504326	0.999990881
21m-63-27F*	**0.880625476**	**0.825360924**	**0.822861822**	0.990520344
21m-65-27F	0.986325434	0.999990787	0.999990787	0.999895439
21m-66-27F	NA	NA	NA	NA
21m-67-27F	0.999999957	0.99999697	0.999996992	0.999966109
21m-68-27F	0.99999981	0.99998327	NA	0.999983149
21m-70-27F	0.999999359	0.999996268	0.999996268	0.999991529
21m-71-27F	0.99985004	0.999995199	0.999995199	0.999771713
21m-72-27F	0.99999655	0.996681307	NA	0.999996955
21m-76-27F	0.999999988	0.999998354	0.999998363	0.999996232
21m-81-27F	0.999999928	0.999987794	0.999986911	0.999978358
21m-82-27F	NA	0.999832758	0.999832758	0.999819939
21m-83-27F	0.999579834	0.999993462	0.999993667	0.99995589
21m-84-27F	0.999999816	0.999986865	0.999986865	0.999983076
21m-85-27F	**0.850149712**	0.999982378	0.999981951	0.999993487

21m-86-27F	NA	0.99682474	0.99682474	0.999970848
21m-87-27F	0.999999946	0.999459162	0.999459488	0.999991363
21m-91-27F	0.999999143	0.999733801	0.999734327	0.999997782
21m-92-27F	0.999999993	0.99999712	0.999997098	0.999997873
21m-94-27F	0.999965615	0.999878505	0.999876502	0.999574867
6m-02-27F	0.948124255	0.999886989	0.999886989	0.999973135
6m-04-27F	0.9999997	0.999993308	0.999993308	0.999993803
6m-05-27F	0.999999809	0.999973188	0.999973188	0.999969356
6m-06-27F	0.999999945	0.99999637	0.999996606	0.999994691
6m-09-27F	0.999999932	0.999996944	0.999996945	0.99999668
6m-10-27F	0.999700326	0.999044474	0.999044474	0.999762791
6m-13-27F	0.999999965	0.999990493	0.999990493	0.999983703
6m-14-27F	0.999999946	0.999995685	0.999995721	0.999993595
6m-15-27F	0.962483452	0.999985389	0.9999855	0.999964317
6m-16-27F	0.99999992	0.999903105	NA	0.999982484
6m-17-27F	0.999991884	0.999662611	0.999662787	0.999992086
6m-19-27F	0.999999916	0.999959879	0.999959486	0.999940495
6m-22-27F	0.97744411	0.998862541	0.998893941	0.999137677
6m-27-27F	0.999999868	0.955022707	0.955057044	0.99998669
6m-28-27F	0.99874705	0.99855622	NA	0.998620463
6m-29-27F	0.999979883	0.999973291	0.999973291	0.999948297
6m-30-27F	0.9999976	0.999981418	0.999981418	0.999970075
6m-33-27F	0.99999998	0.999990445	0.999990445	0.999990895
6m-34-27F	0.999980535	0.999999831	0.999999125	0.999981344
6m-37-27F	0.999999308	0.999885815	0.999885815	0.99989468
6m-39-27F	0.903636547	0.99999425	0.999994191	0.999996968
6m-40-27F	0.999999319	0.999995966	0.999995966	0.999992703
6m-41-27F	0.999713983	0.999604776	0.999604776	0.999953389
6m-43-27F	0.99996984	0.999970768	0.999970768	0.999972493
6m-44-27F	**0.881376523**	0.999989511	0.999989511	0.999989323
6m-50-27F	NA	0.999867733	0.999867733	0.999823275
6m-51-27F	0.999999964	0.999966258	NA	0.999981791
6m-53-27F	0.999999969	0.999970441	0.999971235	0.999986506
6m-56-27F	**0.829520279**	0.999994788	0.999994778	0.999974311
6m-58-27F	0.999999702	0.999985723	0.999985723	0.999985641
6m-59-27F	0.999999212	0.999929601	0.99992735	0.999918721
6m-63-27F	0.99998016	0.999983384	0.999983384	0.999920585
6m-64-27F	0.999999372	**0.789783151**	**0.789950812**	0.999990593
6m-65-27F	0.999999759	0.999915811	0.999915811	0.99995473
6m-66-27F	**0.877146275**	**0.845708551**	**0.845780711**	0.999983645
6m-70-27F	0.999999199	0.999989507	0.999989513	0.999999429
6m-74-27F	0.999994474	0.997338006	NA	0.999741203

6m-75-27F	0.99998019	0.999686327	0.999686327	0.996771341
6m-77-27F	0.999998362	0.999985919	0.999986018	0.999986266
6m-79-27F	0.998191658	0.999811502	0.999811502	0.999399639
6m-81-27F	0.999999983	0.999999171	0.999999173	0.999997648
6m-83-27F	NA	0.999699594	0.999699594	0.99972764
6m-84-27F	0.999999864	0.99999772	NA	0.999997007
6m-85-27F	0.99999995	0.999989398	0.999989355	0.999996201
6m-87-27F	0.999999702	NA	NA	0.999977672
6m-91-27F	0.99999992	NA	NA	0.999987023
6m-94-27F	0.999931904	0.999929312	0.999929312	0.999935375

* Clone names and the corresponding correlations below 0.9 are in bold

Table S5. Correct and incorrect OTUs predicted by distribution-based clustering on simulated data

Dataset	**Correct OTUs**	**Incorrect OTUs**
Constant error rate[1]	40	3
Variable error rate[2]	40	157

[1] Error rate generated from a geometric distribution was 0.9 for 9 libraries
[2] Error rate was 0.8 across 3 libraries and 0.9 across 6 libraries