UPARSE: highly accurate OTU sequences from microbial amplicon reads

Robert C. Edgar, Independent Investigator, Tiburon, California, USA.
robert@drive5.com

Supplementary Information

Nature Methods: doi:10.1038/nmeth.2604

Supplementary Note 1. Data and results for OTU analysis

SN1.1 Most QIIME OTUs are spurious.

The number of OTUs reported by QIIME on the 454 reads ranges from 1,247 (Stag1P) to 3,647 (Stag2P) (see Table SN1.4), many more than would be expected on a mock community with 21 species. This is consistent with the findings of Bokulich *et al.*[1], who obtained large numbers of OTUs in a preliminary QIIME analysis of the MiSeq reads, reducing the number to 206 only after applying a post-hoc OTU size cutoff (their "secondary filtration" parameter *c*). The lack of chimera filtering is a significant problem, as chimeras comprise an estimated 25% to 67% of the QIIME OTUs (Table SN1.4). To investigate further, I compared OTU sequences to the mock community reference database over a sliding window of 150nt. Results for Even1P are shown in Fig. SN1.1, similar results were seen on the other sets. Each window was separately aligned to the reference database to find the best match in order to suppress reduced identities due to chimeric crossovers. This shows that most QIIME OTUs have high identity (mean 98%) with the reference database at the beginning of the sequence, falling to lower identities towards the end of the sequence (mean 87% in the window for positions 400-550). This shows that the large number of OTUs is primarily due to high read error rates, especially towards the end of the sequence, as quality tends to drop as the position increases (Fig. SN1.1, lower plot, see also Fig. SN3.1).

To investigate whether these problems can be mitigated by improved quality and chimera filtering, I ran the QIIME *pick_otus.py* script on the quality filtered and trimmed reads produced by UPARSE, both with and without a preprocessing step using UCHIME *de novo*[2] to filter chimeras. Results are shown in Table SN1.5, which demonstrate that UPARSE quality filtering and trimming and chimera filtering with UCHIME *de novo* followed by *pick_otus.py* gave results comparable with mothur, with between 41 and 213 OTUs per sample of which approximately half are chimeric.

SN1.2 Results on fugal mock communities.

Two fungal mock communities (here called Even and Staggered) based on 11 species were created by Ihrmark *et al*.[6] and sequenced by 454 GS FLX Titanium. Reads for three primers designated ITS1f, gITS7 and fITS9 were obtained from the NCBI Short Read Archive

(accession SRA052087) and a reference database was created using the Genbank sequences specified in Irhmark *et al*.'s Table 3.

Results for UPARSE and QIIME are summarized in Table SN1.3. In this case, I used the QIIME defaults except that UCHIME *de novo* was used to filter chimeras before clustering as this is a reasonable strategy that might be followed by a knowledgeable user of QIIME. Comparison with mothur was not possible because it requires a reference multiple alignment, and none is provided for the ITS region. I encountered technical difficulties with AmpliconNoise on these datasets which were not resolved by the submission deadline.

QIIME created from 298 to 1,863 OTUs, none of which had similarity ≥97% with a known biological sequence.

UPARSE created from 30 to 167 OTUs. Sensitivity to detectable species (i.e., species with at least one read having 3% or fewer errors) ranges from 50% on Stag3.ITS1f to 100% on Even.gITS7. The low sensitivity on Stag3.ITS1f is explained by the fact that there are only four detectable species from the designed community, two of which were not recovered due to very low coverage: *Rhizina undulata* (present in two singleton reads) and *Fusarium poae* (one singleton read).

A large fraction of the UPARSE OTUs are classified as Contaminant or Other. This is at least partly explained by the experimental procedures used to create the community (Björn Lindahl, personal communication), which unfortunately complicate interpretation of the results. Many of the community species were obtained from field-collected fruit bodies, so some contaminants were expected due to fungal material (e.g., spores) from other species in the environment. Manual analysis of selected OTUs classified as Other align well to known fungi in the conserved 28S and 5.8S regions, suggesting that they are due to novel species rather unfiltered errors or chimeras, though this cannot be established definitively (Björn Lindahl, personal communication). If the Other OTUs are assumed to be novel contaminants, then the results are consistent with high accuracy for UPARSE on these datasets.

SN1.3 UPARSE detection of chimeras with >2 segments.

Table SN1.6 summarizes chimeras with >2 segments reported by UPARSE-REF in the mothur and AmpliconNoise OTUs by comparison with the mock community reference database. See Fig. SN1.2 for an example alignment.

UPARSE-REF is not designed to maximize chimera detection accuracy as it has been defined previously in the literature. With the parameters used in UPARSE-OTU, any chimeric model is enough to cause a read to be discarded, regardless of how many differences are implied. This design probably results in many more false positives by the standards of previous benchmarks, but regardless UPARSE is able to generate highly accurate OTUs without a large cost in sensitivity. To resolve the apparent paradox, consider the characteristics desired of a chimera detection method in the context of OTU construction compared to the typical bioinformatics goal of achieving a compromise between high sensitivity and high specificity. The benchmarking methods introduced by Haas *et al.*[3] and further developed by Edgar *et al.*[2] (*mea culpa*) strongly emphasized sensitivity to low-divergence chimeras, defined as chimeras formed from parents with high similarity, and rigorous suppression of false positives, defined as any biological sequence classified as chimeric by comparison with large databases of other known biological sequences. However, if the parents of a 2-segment chimera are highly similar having, say, 97% identity, then the chimera must be <3% diverged from its closest parent (*C*) because at least half of the sequence must be identical to *C*. Such a chimera will usually be merged into the OTU for *C*, in which case it would be harmless if not detected by a chimera filtering stage in a pipeline. Similarly, false positives are harmless if they are reads with errors that produce a weak chimeric signal that is less than 3% different from an existing OTU, and some false positives due to genuinely novel biological sequences are tolerable if this enables us to achieve a large reduction in the OTU error rate. For our purposes here, it is more important to minimize false *negatives* with *high* divergence ≥3% from the closest parent sequence as those are the misclassifications that will produce spurious OTUs. Sensitivity of, say, 99% sounds good, but it is better to think of it as a 1% false negative rate. With millions of reads, a problem of the order of 1% can induce many bad OTUs. Here,

achieving acceptably low error rates in the final OTUs is a major challenge. Sensitivity to low-abundance biological sequences is unimportant unless spurious OTUs can be suppressed to the point where the signal to noise ratio is sufficient to allow robust biological inferences in downstream analysis.

SN1.4 Results with variants of UPARSE.

Table SN1.9 reports results with variants of UPARSE designed to show the essential contribution made by each step in the pipeline. For Illumina, only the recommended "m" sets are shown (merged pairs allowing mismatches) in an effort to make the tables smaller and more easily comparable; similar results are obtained with unpaired reads. Table SN1.9a shows results with the default pipeline for comparison. SN1.9b shows results without length trimming. SN1.9c shows results with UCHIME *de novo* followed by UCLUST[7] used in place of UPARSE-OTU. Chimera-filtered reads were sorted in order of decreasing abundance before running UCLUST. SN1.9d shows results if singletons are retained. In all cases, accuracy is significantly degraded, with increased numbers of Chimeric and Other OTUs. This is particularly striking when singletons are retained: the number of OTUs increases from ~20 to between 53 and 1,354, many or most of which are classified as Chimeric and Other. The number of Contaminants also increases to as many as 95 (Stag2m). At first glance, this might appear to support an argument that retaining singletons can greatly improve sensitivity to low-abundance species. However, closer examination of the MEGABLAST hits calls this conclusion into question. Unlike other datasets, where most contaminants are aligned to sequences obtained from isolate species, most of the hits are to environmental 16S sequences, which may be artifacts of previous experiments, which are reproduced surprisingly often[3]. This underscores the difficulty of reliably identifying contaminants and reminds us that results must be interpreted with caution.

Figure SN1.1 Local identity of QIIME OTUs correlates with read error rate.

Mean identity of QIIME Even1P OTUs with the 16S mock community reference database is shown computed over a sliding window of length 150nt (upper red line). Bars show the standard deviation. The lower line (blue) shows the mean probability that the base call is correct as a function of position in the read, as predicted by quality scores in the FASTQ records. These results show that most QIIME OTUs are derived from reads of expected species in the mock community rather than contaminants. The lack of quality trimming and chimera filtering produces large numbers of OTUs that are >3% diverged from the original biological sequences.

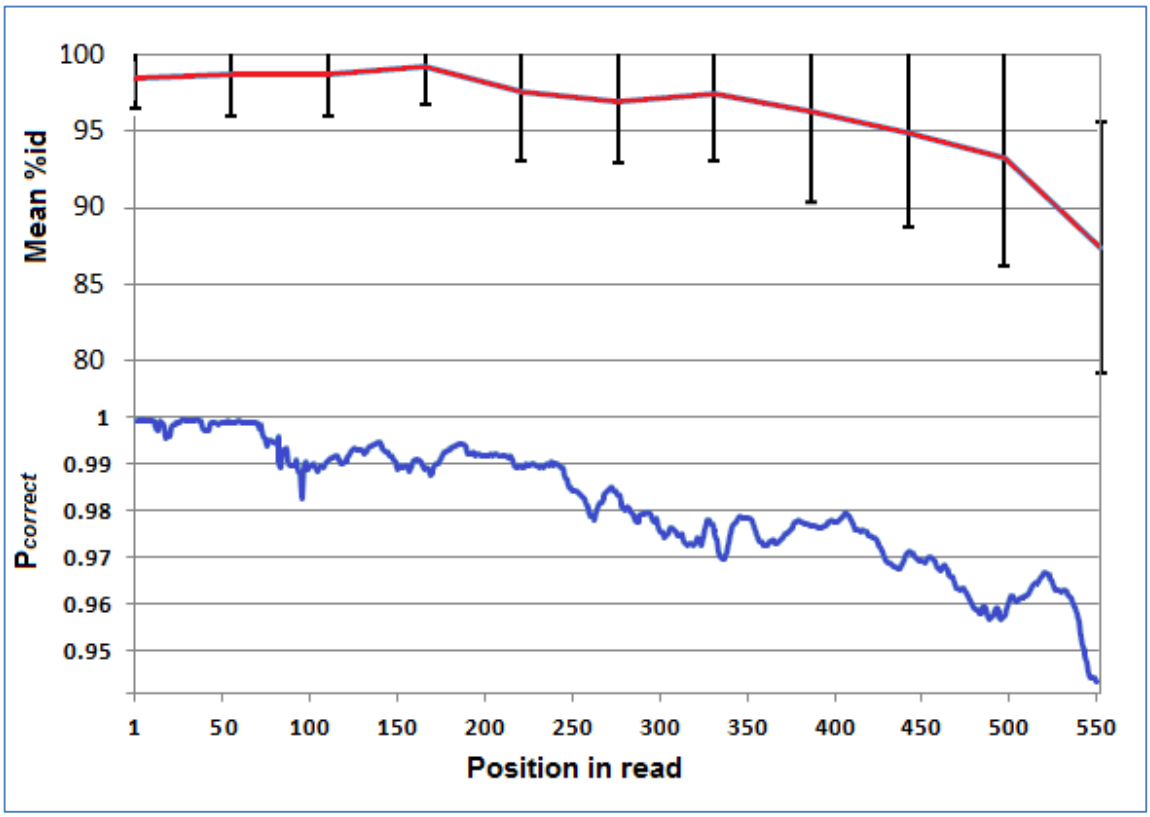
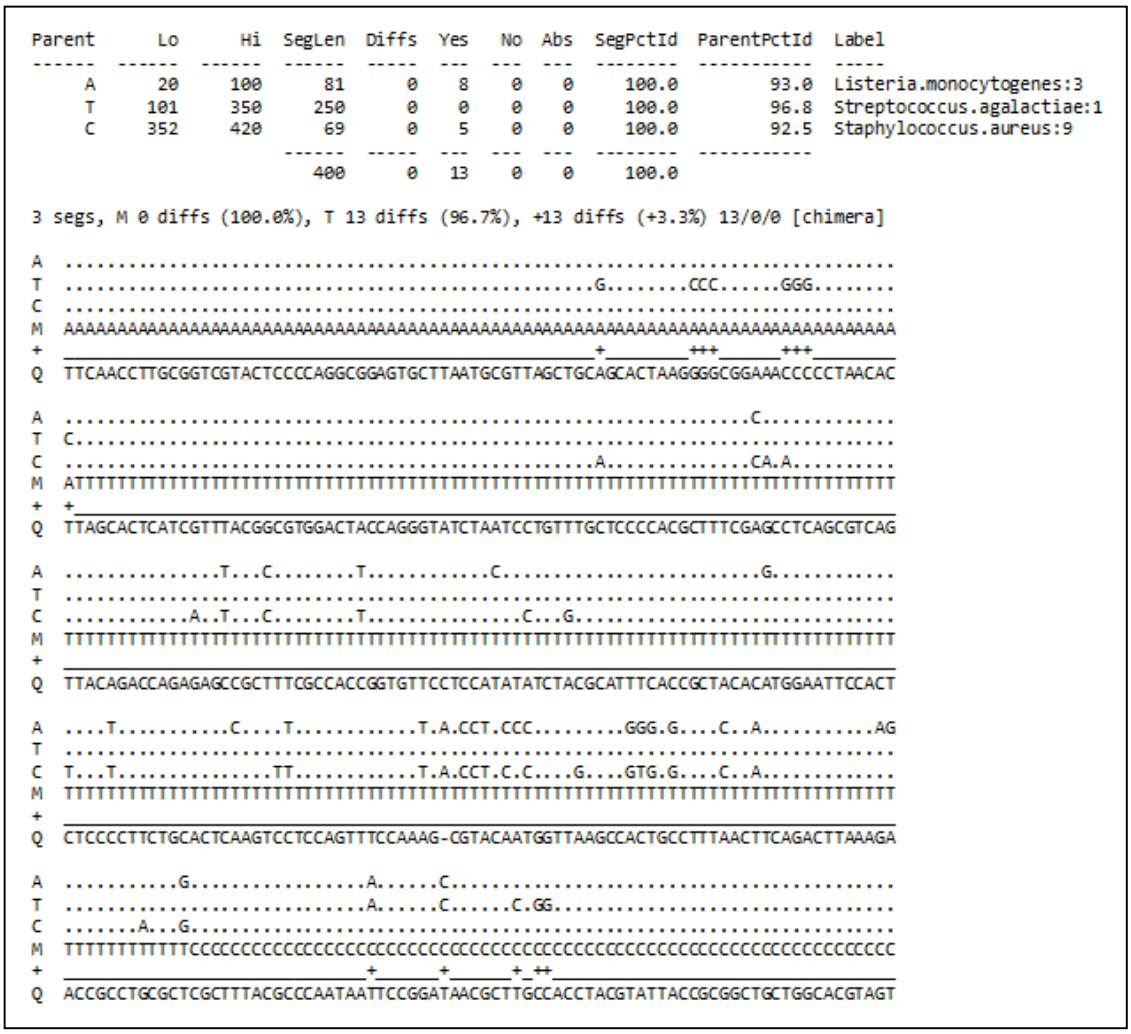


Figure SN1.2. UPARSE-REF alignment of a 3-segment chimera to the reference database.

This figure shows the alignment generated by UPARSE-REF of an AmpliconNoise OTU sequence from Even1P to the reference database. The OTU sequence is identical to reads FO09O1002JK77H and FO09O1002JW207. The OTU sequence is Rows are A, T and C for the three parent sequences (T is also the closest sequence in the reference database), M representing the model, which has letters A, T and C indicating the parent segment, + showing differences that support the model, and Q showing the query sequence. In rows A, T and C, dots indicate a letter that is identical with the query sequence. The alignment shows that the model constructed from three segments is identical to the query, while the closest reference sequence is 3.3% different. This chimera will therefore create a spurious OTU if it is not detected.



```
Parent      Lo      Hi  SegLen  Diffs  Yes   No  Abs  SegPctId  ParentPctId  Label
------  ------  ------  ------  -----  ---  ---  ---  --------  -----------  -----
     A      20     100      81      0    8    0    0     100.0         93.0  Listeria.monocytogenes:3
     T     101     350     250      0    0    0    0     100.0         96.8  Streptococcus.agalactiae:1
     C     352     420      69      0    5    0    0     100.0         92.5  Staphylococcus.aureus:9
                   ------  -----  ---  ---  ---  --------  -----------
                      400      0   13    0    0     100.0

3 segs, M 0 diffs (100.0%), T 13 diffs (96.7%), +13 diffs (+3.3%) 13/0/0 [chimera]

A  ..........................................................................
T  ..................................................G........CCC......GGG........
C  ..........................................................................
M  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+  ____________________________________________________+________+++______+++________
Q  TTCAACCTTGCGGTCGTACTCCCCAGGCGGAGTGCTTAATGCGTTAGCTGCAGCACTAAGGGGCGGAAACCCCCTAACAC

A  ..............................................................C.............
T  C.........................................................................
C  ..................................................A..............CA.A..........
M  ATTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
+  +_____________________________________________________________________________
Q  TTAGCACTCATCGTTTACGGCGTGGACTACCAGGGTATCTAATCCTGTTTGCTCCCCACGCTTTCGAGCCTCAGCGTCAG

A  ...............T...C........T............C.........................G............
T  .........................................................................
C  ............A..T...C........T................C...G.....................................
M  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
+  _____________________________________________________________________________
Q  TTACAGACCAGAGAGCCGCTTTCGCCACCGGTGTTCCTCCATATATCTACGCATTTCACCGCTACACATGGAATTCCACT

A  ....T...........C....T............T.A.CCT.CCC.........GGG.G....C..A...........AG
T  .........................................................................
C  T...T...............TT............T.A.CCT.C.C....G....GTG.G....C..A.............
M  TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
+  _____________________________________________________________________________
Q  CTCCCCTTCTGCACTCAAGTCCTCCAGTTTCCAAAG-CGTACAATGGTTAAGCCACTGCCTTTAACTTCAGACTTAAAGA

A  ...........G.................A......C.....................................
T  ............................A......C......C.GG.............................
C  .......A...G.....................................................................
M  TTTTTTTTTTTTCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
+  ____________________________+_____+_____+_++_____________________________________
Q  ACCGCCTGCGCTCGCTTTACGCCCAATAATTCCGGATAACGCTTGCCACCTACGTATTACCGCGGCTGCTGGCACGTAGT
```

Table SN1.1 16S mock datasets.

This table shows details and nomenclature for 16S mock community reads used in this study. SRR is the NCBI Short Read Archive accession. Primers gives the primers, Reads the number of unfiltered 454 reads, Pairs is the number of unfiltered Illumina read pairs. Illumina datasets are from Bokulich *et al.*[1], deposited in the QIIME database[8].

454 datasets

Run	Community	SRR	Primers	Reads
Even1P	Even	SRR053818	V3-V5	37,377
Even2P	Even	SRR072220	V3-V5	14,779
Even3P	Even	SRR072239	V3-V5	13,863
Stag1P	Staggered	SRR072221	V3-V5	13,287
Stag2P	Staggered	SRR072223	V3-V5	29,050
Stag3P	Staggered	SRR072237	V3-V5	8,964

Illumina paired reads

Replicate	Community	Primers	Pairs
Even1	Even	V4	1,520,374
Even2	Even	V4	1,644,911
Stag1	Staggered	V4	1,857,075
Stag2	Staggered	V4	2,576,656

Illumina datasets

Sets	Description
Even1F .. Stag2F	Forward reads only.
Even1R .. Stag2R	Reverse reads only.
Even1m .. Stag2m	Merged pairs, allowing mismatches.
Even1x .. Stag2x	Merged pairs, disallowing mismatches.
AllF	All forward reads Even1F+Even2F+Stag1F+Stag2F.

Table SN1.2. Read abundances per species in 16S mock communities.

This table gives the number of reads per species for the 16S datasets. A read was assigned to a species if it had a global alignment with ≥97% identity to a reference sequence. In the case of Illumina (lower table), only forward reads are shown as these have lower error rates than reverse reads, and merging of pairs may lose some pairs that fail to overlap.

454 datasets

Species	Even1P	Even2P	Even3P	Stag1P	Stag2P	Stag3P
Streptococcus.mutans	451(4.4%)	62(2.8%)	60(2.8%)	1393(49.0%)	2779(54.7%)	507(67.3%)
Deinococcus.radiodurans	756(7.3%)	1503(67.2%)	1353(64.1%)	1(0.0%)	4(0.1%)	0
Staphylococcus.epidermidis	635(6.2%)	103(4.6%)	82(3.9%)	956(33.6%)	1350(26.6%)	129(17.1%)
Bacteroides.vulgatus	2248(21.9%)	55(2.5%)	96(4.5%)	0	4(0.1%)	1(0.1%)
Staphylococcus.aureus	1968(19.1%)	64(2.9%)	60(2.8%)	110(3.9%)	173(3.4%)	17(2.3%)
Acinetobacter.baumannii	1890(18.4%)	101(4.5%)	147(7.0%)	10(0.4%)	34(0.7%)	2(0.3%)
Propionibacterium.acnes	445(4.3%)	144(6.4%)	112(5.3%)	13(0.5%)	17(0.3%)	1(0.1%)
Neisseria.meningitidis	560(5.4%)	33(1.5%)	56(2.7%)	9(0.3%)	17(0.3%)	3(0.4%)
Rhodobacter.sphaeroides	26(0.3%)	22(1.0%)	12(0.6%)	180(6.3%)	364(7.2%)	64(8.5%)
Clostridium.beijerinckii	364(3.5%)	32(1.4%)	17(0.8%)	37(1.3%)	80(1.6%)	5(0.7%)
Streptococcus.pneumoniae	345(3.4%)	30(1.3%)	33(1.6%)	0	1(0.0%)	0
Escherichia.coli	51(0.5%)	6(0.3%)	5(0.2%)	98(3.4%)	178(3.5%)	11(1.5%)
Streptococcus.agalactiae	119(1.2%)	3(0.1%)	4(0.2%)	21(0.7%)	40(0.8%)	7(0.9%)
Actinomyces.odontolyticus	61(0.6%)	53(2.4%)	40(1.9%)	0	0	1(0.1%)
Enterococcus.faecalis	101(1.0%)	11(0.5%)	10(0.5%)	0	0	0
Pseudomonas.aeruginosa	79(0.8%)	7(0.3%)	6(0.3%)	3(0.1%)	19(0.4%)	2(0.3%)
Lactobacillus.gasseri	77(0.7%)	3(0.1%)	6(0.3%)	3(0.1%)	0	1(0.1%)
Bacillus.cereus	62(0.6%)	4(0.2%)	3(0.1%)	8(0.3%)	11(0.2%)	2(0.3%)
Listeria.monocytogenes	49(0.5%)	0	8(0.4%)	1(0.0%)	2(0.0%)	0
Methanobrevibacter.smithii	0	0	0	1(0.0%)	4(0.1%)	0

Table SN1.3. Results on fungal mock communities.

With QIIME, all OTUs were classified as "Other". Mothur was not able to process this set due the lack of a curated reference multiple alignment, and AmpliconNoise was not included owing to unresolved technical difficulties running the software.

UPARSE

Set	Reads	OTUs	Perfect	Good	Noisy	Chimeric	Contaminant	Other
Even.fITS9	$1.2x10^4$	42	7	1	0	0	32	2
Even.gITS7	$7.8x10^4$	167	8	3	0	0	99	57
Even.ITS1f	$3.0x10^4$	157	8	1	0	0	111	37
Stag.fITS9	$1.6x10^4$	30	5	0	0	0	17	8
Stag.gITS7	$8.2x10^4$	182	5	3	1	0	130	43
Stag.ITS1f	$8.2x10^4$	86	2	0	0	0	73	11

QIIME

Set	Reads	OTUs	Perfect	Good	Noisy	Chimeric	Contaminant	Other
Even.fITS9	$1.2x10^4$	371	0	0	0	0	0	371
Even.gITS7	$7.8x10^4$	1,863	0	0	0	0	0	1,863
Even.ITS1f	$3.0x10^4$	1,159	0	0	0	0	0	1,159
Stag.fITS9	$1.6x10^4$	298	0	0	0	0	0	298
Stag.gITS7	$8.2x10^4$	1,860	0	0	0	0	0	1,860
Stag.ITS1f	$8.2x10^4$	657	0	0	0	0	0	657

Table SN1.4. Results on 16S mock datasets.

Columns are: Set=dataset (see Table S1), OTUs=number of OTUs, Perfect, Good, Noisy, Cont[aminant] and Chim[eric] are the numbers of OTUs in each category. See Online Methods for definitions of categories.

UPARSE 454

Set	OTUs	Perfect	Good	Noisy	Cont.	Chim.	Other
Even1P	22	7	13	1	0	1	0
Even2P	20	14	5	0	1	0	0
Even3P	19	14	5	0	0	0	0
Stag1P	16	9	6	0	1	0	0
Stag2P	20	10	8	1	1	0	0
Stag3P	12	6	6	0	0	0	0

AmpliconNoise 454

Set	OTUs	Perfect	Good	Noisy	Cont.	Chim.	Other
Even1P	75	16	4	0	1	48	6
Even2P	49	18	1	1	2	25	2
Even3P	54	18	1	1	2	29	3
Stag1P	26	16	1	2	2	4	1
Stag2P	40	17	2	0	1	18	2
Stag3P	28	16	1	0	1	10	0

mothur 454

Set	OTUs	Perfect	Good	Noisy	Cont.	Chim.	Other
Even1P	141	13	6	9	0	100	13
Even2P	53	18	0	2	0	28	5
Even3P	64	16	0	4	0	35	9
Stag1P	44	12	2	4	0	22	4
Stag2P	71	14	0	10	0	36	11
Stag3P	34	13	2	3	0	14	2

QIIME 454

Set	OTUs	Perfect	Good	Noisy	Cont.	Chim.	Other
Even1P	2518	4	16	9	0	1681	808
Even2P	2257	3	14	4	0	974	1262
Even3P	2160	5	11	8	0	900	1236
Stag1P	1247	6	8	11	0	416	806
Stag2P	3647	7	9	11	0	963	2657
Stag3P	1900	2	7	18	0	437	1436

UPARSE Illumina

m=merged paired reads (recommended), x=merged with mismatches excluded, F=forward only, R=reverse only.

Set	OTUs	Perfect	Good	Noisy	Chim.	Cont.	Other
Even1m	26	20	0	0	0	6	0
Even2m	25	20	0	0	0	5	0
Stag1m	26	20	0	0	0	6	0
Stag2m	24	20	0	0	0	4	0
Even1x	23	20	0	0	0	3	0
Even2x	23	20	0	0	0	3	0
Stag1x	22	19	0	0	0	3	0
Stag2x	23	19	0	0	0	4	0
Even1F	15	13	0	1	1	0	0
Even2F	15	13	0	1	1	0	0
Stag1F	13	11	0	0	0	2	0
Stag2F	12	11	0	0	0	1	0
Even1R	19	19	0	0	0	0	0
Even2R	19	19	0	0	0	0	0
Stag1R	15	15	0	0	0	0	0
Stag2R	18	13	2	0	0	1	2

AllF dataset

Pooled Illumina dataset with all forward reads.

Method	OTUs	Perfect	Good	Noisy	Chime.	Cont.	Other
UPARSE	19	17	0	0	0	2	0
QIIME	206	24	34	31	93	0	24

Table SN1.5. 16S 454 mock community results with variants of QIIME.

This table shows evaluations for OTUs produced by the QIIME *pick_otus.py* script from reads that were quality-filtered by USEARCH. Methods are: UQ, using USEARCH quality-filtered reads, and Uch+UQ where UCHIME *de novo* was run on the trimmed reads before running *pick_otus.py*. Columns are: Nr OTUs, number of OTUs, *V*=100% and *V*<97% where *V* is identity with the closest reference sequence, and Chimeras gives the number of chimeras reported by UPARSE-REF and UCHIME respectively using the mock community reference database.

Set	Method	Nr OTUs	V=100%	V<97%	Chimeras
Even1P	UQ	413	3	364	306 / 348
	Uch+UQ	213	8	186	143 / 170
Even2P	UQ	193	0	156	148 / 162
	Uch+UQ	84	14	65	60 / 62
Even3P	UQ	189	0	150	143 / 158
	Uch+UQ	86	13	65	55 / 62
Stag1P	UQ	91	1	65	59 / 65
	Uch+UQ	51	9	34	25 / 32
Stag2P	UQ	149	1	106	103 / 111
	Uch+UQ	92	12	65	57 / 65
Stag3P	UQ	62	2	39	39 / 43
	Uch+UQ	41	7	23	20 / 22

Table SN1.6. Chimeras with >2 segments in mothur and AmpliconNoise OTUs.

This table shows the number of chimeras with number of segments $m>2$ found by UPARSE-REF in the mothur and AmpliconNoise (AN) OTUs using the 16S mock community reference database. Here, d is the number of differences (mismatches plus gaps) between the OTU sequence and the UPARSE model. If $d=0$, the chimeric model is identical to the OTU so is almost certainly correct, and if $d \leq 3$ then the model is probably correct since the breakpoint penalty $b=3$ ensures that the read must have at least $3\times(m-1)$ more differences with the closest parent sequence than with the chimeric model. In a read of length 250, a read with a model with $m=3$ and $d=3$ must be ~3.5% diverged from the closest reference sequence but is only ~1% diverged from the chimeric model, making the chimera prediction strongly credible. UCHIME, the most sensitive previously published methods, reports less than half of the $d \leq 3$ chimeras using the mock community reference database (last column), showing the importance of improved chimera filtering.

mothur	$m=3$		$m=4$		$m=5$		
	$d \leq 3$	$d = 0$	$d \leq 3$	$d = 0$	$d \leq 3$	$d = 0$	UCHIME
Even1P	51	26	3	2	1	0	20
Even2P	10	9	1	1	0	0	3
Even3P	10	6	1	0	0	0	3
Stag1P	12	9	1	1	0	0	4
Stag2P	22	15	1	1	0	0	8
Stag3P	8	4	0	0	0	0	5

AN	$m=3$		$m=4$		$m=5$		
	$d \leq 3$	$d = 0$	$d \leq 3$	$d = 0$	$d \leq 3$	$d = 0$	UCHIME
Even1P	12	7	2	0	0	0	5
Even2P	5	2	0	0	0	0	2
Even3P	4	2	0	0	0	0	1
Stag1P	1	1	0	0	0	0	0
Stag2P	6	3	0	0	0	0	2
Stag3P	2	1	0	0	0	0	0

Illumina datasets

Species	Even1F	Even2F	Stag1F	Stag2F
Staphylococcus.aureus	652811(48.8%)	703126(50.1%)	1154884(69.2%)	1532791(68.1%)
Acinetobacter.baumannii	290480(21.7%)	264653(18.8%)	15840(0.9%)	22647(1.0%)
Streptococcus.mutans	18171(1.4%)	19825(1.4%)	129549(7.8%)	183274(8.1%)
Rhodobacter.sphaeroides	13648(1.0%)	13017(0.9%)	108835(6.5%)	151485(6.7%)
Methanobrevibacter.smithii	6561(0.5%)	7700(0.5%)	111533(6.7%)	147893(6.6%)
Deinococcus.radiodurans	81553(6.1%)	101711(7.2%)	975(0.1%)	1433(0.1%)
Escherichia.coli	8686(0.6%)	9938(0.7%)	63781(3.8%)	95444(4.2%)
Pseudomonas.aeruginosa	38165(2.9%)	40374(2.9%)	27416(1.6%)	39163(1.7%)
Clostridium.beijerinckii	31398(2.3%)	34746(2.5%)	26083(1.6%)	35943(1.6%)
Bacteroides.vulgatus	42576(3.2%)	47736(3.4%)	343(0.0%)	498(0.0%)
Bacillus.cereus	18322(1.4%)	19591(1.4%)	13911(0.8%)	19965(0.9%)
Listeria.monocytogenes	31286(2.3%)	34335(2.4%)	1976(0.1%)	3378(0.1%)
Helicobacter.pylori	31594(2.4%)	33645(2.4%)	2245(0.1%)	3062(0.1%)
Streptococcus.agalactiae	16218(1.2%)	16004(1.1%)	9807(0.6%)	14066(0.6%)
Streptococcus.pneumoniae	24206(1.8%)	23107(1.6%)	73(0.0%)	125(0.0%)
Actinomyces.odontolyticus	14062(1.1%)	15934(1.1%)	87(0.0%)	103(0.0%)
Enterococcus.faecalis	12380(0.9%)	13642(1.0%)	162(0.0%)	244(0.0%)
Neisseria.meningitidis	2427(0.2%)	3050(0.2%)	258(0.0%)	367(0.0%)
Lactobacillus.gasseri	1815(0.1%)	2435(0.2%)	104(0.0%)	198(0.0%)
Propionibacterium.acnes	127(0.0%)	181(0.0%)	13(0.0%)	11(0.0%)

Table SN1.7. 454 pipeline statistics.

This table shows the number of sequences at each stage in the 454 pipelines. Den. = after denoising, -Ch = after chimera filtering, Qual. = after quality filtering, OTUs = number of OTUs. The Uch+UQ variant of QIIME was used (see Table SN1.5) as this gave the best results; this uses UPARSE quality filtering following by UCHIME *de novo* before clustering by UCLUST.

		AmpliconNoise			mothur			QIIME			UPARSE	
Set	**Reads**	**Den.**	**-Ch**	**OTUs**	**Den.**	**-Ch**	**OTUs**	**Qual.**	**-Ch**	**OTUs**	**Qual.**	**OTUs**
Even1P	37,377	851	86	75	3,518	492	141	29,503	3,050	213	29,503	19
Even2P	14,779	409	55	49	2,166	237	53	9,262	1,780	84	9,262	18
Even3P	13,863	396	71	54	1,841	250	64	7,092	1,377	86	7,092	16
Stag1P	13,287	135	36	26	1,401	167	44	7,193	1,311	51	7,193	14
Stag2P	29,050	236	54	40	2,561	317	71	13,759	2,391	92	13,759	16
Stag3P	8,964	124	36	28	802	141	34	2,434	541	41	2,434	11

Table SN1.8. Illumina pipeline statistics.

This table shows the number of sequences at each stage in the Illumina pipelines for the forward reads. Qual. = reads after quality filtering, OTUs (before c) = number of OTUs before size cutoff applied, OTUs (after c) = number of OTUs after size cutoff. For QIIME, all samples are pooled into a single set of reads (AllF), following the methodology of Bokulich *et al.*[1].

QIIME

Set	Reads	Qual	OTUs (before c)	OTUs (after c)
AllF	7,599,016	7,179,038	64,228	206

UPARSE

Set	Reads	Qual.	OTUs
AllF	7,599,016	7,550,545	19
Even1F	1,520,374	1,512,253	14
Even2F	1,644,911	1,634,886	15
Stag1F	1,857,075	1,845,351	13
Stag2F	2,576,656	2,558,055	12

Table SN1.9. Results with variants of UPARSE.

Columns are: Set=dataset (see Table S1), OTUs=number of OTUs, Perfect, Good, Noisy, Cont[aminant] and Chim[eric] are the numbers of OTUs in each category. See Online Methods for definitions of categories and section S7 for discussion.

12a. Defaults.

Set	OTUs	Perfect	Good	Noisy	Chim.	Cont.	Other
Even1P	19	7	10	1	1	0	0
Even2P	17	12	4	1	0	0	0
Even3P	16	12	4	0	0	0	0
Stag1P	13	7	6	0	0	0	0
Stag2P	15	9	5	0	0	1	0
Stag3P	11	6	5	0	0	0	0
Even1m	25	18	1	1	0	5	0
Even2m	22	18	1	0	0	3	0
Stag1m	24	18	1	0	0	4	1
Stag2m	21	17	1	0	0	3	0

12b. Without length trimming.

Set	OTUs	Perfect	Good	Noisy	Chim.	Cont.	Other
Even1P	30	7	11	1	4	3	4
Even2P	25	13	6	0	1	0	5
Even3P	23	10	5	2	1	1	4
Stag1P	19	7	4	2	1	1	4
Stag2P	25	7	4	3	4	3	4
Stag3P	18	6	5	3	1	2	1
Even1m	51	21	0	2	0	9	19
Even2m	54	21	1	1	0	8	23
Stag1m	66	21	0	0	0	9	36
Stag2m	156	21	1	20	9	14	91

12c. UCHIME and UCLUST instead of UPARSE-OTU

Set	OTUs	Perfect	Good	Noisy	Chim.	Cont.	Other
Even1P	69	8	12	10	34	0	5
Even2P	27	14	4	0	9	0	0
Even3P	22	13	5	0	4	0	0
Stag1P	26	9	6	0	11	0	0
Stag2P	45	9	9	1	25	1	0
Stag3P	9	5	4	0	0	0	0
Even1m	162	20	0	1	136	5	0
Even2m	180	20	0	1	156	3	0
Stag1m	96	20	0	0	71	4	1
Stag2m	117	19	0	0	95	3	0

12d. Singletons retained

Set	OTUs	Perfect	Good	Noisy	Chim.	Cont.	Other
Even1P	69	7	12	2	17	7	24
Even2P	76	13	6	3	15	3	36
Even3P	91	10	6	4	19	9	43
Stag1P	53	8	7	4	6	7	21
Stag2P	87	8	6	8	14	5	46
Stag3P	49	6	9	5	7	4	18
Even1m	1008	22	0	137	86	51	712
Even2m	1164	23	4	158	110	70	799
Stag1m	968	21	2	115	48	52	730
Stag2m	1354	21	2	157	72	95	1007

Table SN1.10. SRA accessions for HMP datasets

HMP body site	Short name
L Retroauricular crease	Ear
R Retroauricular crease	Ear
L Antecubital fossa	Elbow
R Antecubital fossa	Elbow
Posterior fornix	Fornix
Attached/Keratinized gingiva	Gingiva
Vaginal introitus	IntVagina
Mid vagina	MidVagina
Buccal mucosa	Mouth
Anterior nares	Nostril
Hard palate	Palate
Saliva	Saliva
Stool	Stool
Subgingival plaque	SubPlaque
Supragingival plaque	SupPlaque
Throat	Throat
Tongue dorsum	Tongue
Palatine Tonsils	Tonsils

SRA accession	Body site
SRS013674	Anterior_nares
SRS013741	Anterior_nares
SRS017971	Anterior_nares
SRS021520	Anterior_nares
SRS022129	Anterior_nares
SRS023950	Anterior_nares
SRS013656	Attached/Keratinized_gingiva
SRS013714	Attached/Keratinized_gingiva
SRS017953	Attached/Keratinized_gingiva
SRS021502	Attached/Keratinized_gingiva
SRS022111	Attached/Keratinized_gingiva
SRS023932	Attached/Keratinized_gingiva
SRS013654	Buccal_mucosa
SRS013711	Buccal_mucosa
SRS017951	Buccal_mucosa
SRS021500	Buccal_mucosa
SRS022109	Buccal_mucosa
SRS023930	Buccal_mucosa
SRS013652	Hard_palate
SRS013708	Hard_palate
SRS017949	Hard_palate
SRS021498	Hard_palate
SRS022107	Hard_palate
SRS023928	Hard_palate
SRS013670	L_Antecubital_fossa
SRS021516	L_Antecubital_fossa

SRA accession	Body site
SRS022125	L_Antecubital_fossa
SRS013666	L_Retroauricular_crease
SRS013727	L_Retroauricular_crease
SRS017963	L_Retroauricular_crease
SRS021512	L_Retroauricular_crease
SRS022121	L_Retroauricular_crease
SRS023942	L_Retroauricular_crease
SRS022133	Mid_vagina
SRS013658	Palatine_Tonsils
SRS013717	Palatine_Tonsils
SRS017955	Palatine_Tonsils
SRS021504	Palatine_Tonsils
SRS023934	Palatine_Tonsils
SRS022135	Posterior_fornix
SRS013672	R_Antecubital_fossa
SRS013738	R_Antecubital_fossa
SRS022127	R_Antecubital_fossa
SRS023948	R_Antecubital_fossa
SRS013668	R_Retroauricular_crease
SRS013732	R_Retroauricular_crease
SRS017965	R_Retroauricular_crease
SRS021514	R_Retroauricular_crease
SRS022123	R_Retroauricular_crease
SRS023944	R_Retroauricular_crease
SRS013646	Saliva
SRS013699	Saliva
SRS017943	Saliva
SRS021492	Saliva
SRS022101	Saliva
SRS023922	Saliva
SRS024036	Saliva
SRS013638	Stool
SRS013687	Stool
SRS021484	Stool
SRS022093	Stool
SRS023914	Stool
SRS024028	Stool
SRS013664	Subgingival_plaque
SRS013729	Subgingival_plaque
SRS017961	Subgingival_plaque
SRS021510	Subgingival_plaque
SRS022119	Subgingival_plaque
SRS023940	Subgingival_plaque
SRS013662	Supragingival_plaque
SRS013723	Supragingival_plaque
SRS017959	Supragingival_plaque
SRS021508	Supragingival_plaque
SRS022117	Supragingival_plaque
SRS023938	Supragingival_plaque
SRS013660	Throat

SRA accession	Body site
SRS013720	Throat
SRS017957	Throat
SRS021506	Throat
SRS022115	Throat
SRS023936	Throat
SRS013650	Tongue_dorsum
SRS013705	Tongue_dorsum
SRS017947	Tongue_dorsum
SRS021496	Tongue_dorsum
SRS022105	Tongue_dorsum
SRS023926	Tongue_dorsum
SRS022131	Vaginal_introitus

Supplementary Note 2. Singletons and errors.

SN2.1 Discussion of singleton sequences

By definition, a singleton has a sequence found exactly once in the reads. Here, a bad read is defined to be a read with at least one error. Bad bases may be due to any experimental artifact, including sequencing, chimeras and PCR point errors.

Let P(e) be the probability that a read contains e errors. Consider first a highly simplified model where P(0) = 1/2, P(1) = 1/2 and P(e>1) = 0. Suppose a given unique amplicon sequence has 100 reads, then this should give ∼50 correct reads and ∼50 bad reads, each with exactly one error. Suppose the read length is 500 and the only possible error is an incorrect base call, then there are 500 positions × 3 possible wrong bases = 1,500 possible incorrect read sequences. Assuming the errors are randomly distributed, then we expect most of the bad reads to be singletons, though per the birthday paradox errors may be duplicated by chance more often than one would naively expect. Notice that we have ∼50 unique sequences that are bad, but only one correct sequence.

More realistically and more generally, suppose the following assumptions hold:

(i) P(0) is large enough that a significant fraction of the reads is correct,

(ii) P(e>0) = 1 – P(0) is large enough that a significant fraction of the reads are bad, i.e. have one or more errors, and

(iii) The probability of a given error being reproduced is small.

Under these assumptions, a unique amplicon sequence Q with N reads will produce ∼N P(0) correct reads (i.e., one unique read with abundance ∼N P(0)), and ∼N P(e>0)

singletons having unique errors. There may also be a few bad reads with abundance >1 due to duplicated errors, but these are rare by assumption (iii). We can picture the bad reads as an "error cloud" composed mostly of singletons surrounding the correct sequence (Fig. SN2.1).

Notice that the abundance of the correct sequence and the number of singleton bad reads derived from Q are both approximately proportional to N, the total number of reads for Q. Therefore, if we increase the number of reads by a factor d, the number of singletons due to errors will also increase by a factor of approximately d.

Now suppose that in addition to some high-abundance amplicons, there are some rare biological sequences, some of which have exactly one read. For the number of correct or nearly correct singletons due to rare amplicons to be comparable with the number of singletons due to error clouds, there would have to be a very large number of low-abundance biological sequences, bearing in mind that the requirement to produce exactly one read is very stringent -- many of them would be expected to produce zero or more than one read due to variations in abundance, read quality etc.

If we use a sequencing technology with a significant error rate, and we observe some high-abundance reads and many singletons, then a large fraction of the singletons almost certainly belong to the error clouds of high-abundance sequences. (Some of the high-abundance reads may themselves have errors, e.g. chimeras, but these will also induce error clouds). These conclusions follow from the observations and assumptions (i), (ii) and (iii), regardless of the possible presence of low-abundance biological sequences.

Table SN2.1 summarizes statistics for singletons and high-abundance reads in the 16S mock datasets and some real datasets from the Human Microbiome Project, which show that some high-abundance reads and many singletons are indeed observed. As expected, singletons are a large fraction of the reads, ranging from 79% to 87% of the unique sequences. There are several read sequences with high abundance; these must have error clouds unless P(e>0) is very close to zero, which is contradicted by error probabilities

according to Phred scores in the reads and by error rates measured on control sequences[4,5]. Only a tiny minority of the singletons in the mock datasets are classified as Perfect, i.e. free of errors. Many are Good, meaning up to 1% errors, but most of these probably belong to error clouds for which a correct sequence with abundance > 1 is also present, meaning that discarding those Good singletons would not degrade sensitivity. Hundreds of other singletons in each sample are Noisy, Chimeric or Other. Chimeric sequences would certainly generate spurious OTUs, since by definition they are >3% diverged from the closest biological sequence. Many of the Noisy sequences would probably generate spurious OTUs, though some will be absorbed into a valid OTU. (While a Noisy sequence has ≤3% errors and will therefore be absorbed if its correct sequence is the centroid, it may fall outside the OTU otherwise). Most of the Other sequences probably have >3% errors (due to sequencing or undetected chimeras) which would also generate spurious OTUs, though it cannot be ruled out that a few are valid. The prediction that large numbers of spurious OTUs would be produced are supported by UPARSE results when singletons are retained (Table SN1.9), where most OTUs are classified as Other and are most likely spurious. A real community may have higher diversity than a mock community. However, if we sequence a real community and find several high-abundance sequences, then it follows from the mock community results that we can expect a comparable large number of spurious singletons due to the error clouds of those sequences, regardless of the possible presence of more low-abundance biological sequences.

Figure SN2.1. Singleton reads and error clouds.

This figure illustrates features expected under assumptions described in section SN2.1. Most unique reads are singletons, and most singletons have at least one error. High-abundance reads are surrounded by an "error cloud" composed mostly of singletons. The number of singletons in a cloud is approximately proportional to the abundance of the correct sequence (larger green dots indicate higher abundance). Most singletons have few errors and can be absorbed into the OTU for their correct sequence. (In the context of UPARSE, this is done in a post-processing step that maps unfiltered reads to OTU sequences). The remainder can be classified as "isolated" and "bad" singletons. Isolated singletons have ≤3% errors and are the only reads for a given taxon; they would generate valid OTUs that will otherwise be missed. Bad singletons have >3% errors and would generate spurious OTUs. Both isolated and bad singletons are expected to be small fractions of the singletons. While a small improvement in sensitivity may be achieved by keeping singletons in order preserve the isolated subset, this may result in an unacceptable increase in spurious OTUs due to bad singletons.

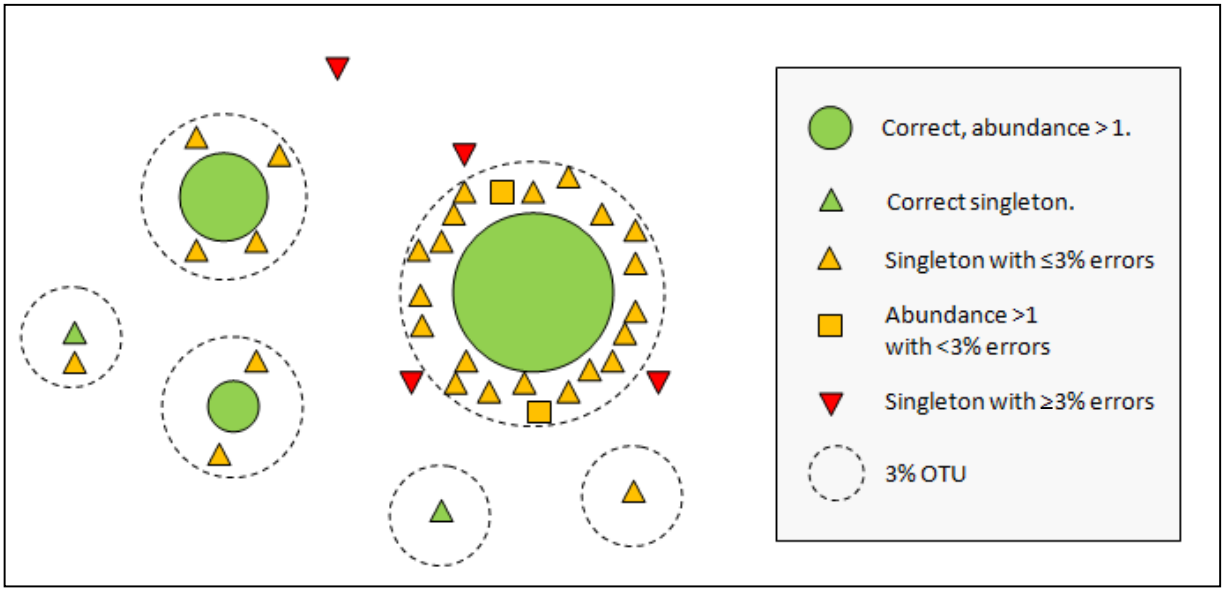


Table SN2.1. Singleton statistics for mock and real samples.

Columns are: Set: dataset; Type: sequencer; SRR: SRA accession; Reads: total number of reads; Ab>100: number of unique read sequences with abundance > 100; Non-singletons: number of non-singleton unique read sequences; Singles: number of singletons; Pct.Sgl.: percentage of unique read sequences that are singletons; Good, Noisy, Chimera and Other: classifications defined in Online Methods.

Set	Ab>100	Non-singles	Singles	Pct.Sgl.	Perfect	Good	Noisy	Chimera	Other
Even1P	36	1,452	5,600	79%	0	1,409	153	834	3,204
Even2P	17	617	3,628	85%	0	1,549	119	482	1,478
Even3P	15	526	3,535	87%	4	1,440	112	483	1,496
Stag1P	9	476	2,370	83%	2	958	65	282	1,063
Stag2P	15	1,100	4,378	80%	3	1,466	135	571	2,203
Stag3P	5	254	1,695	87%	1	702	49	186	757
Even1F	12	2,128	8,113	79%	4	6,553	249	392	915
Even2F	13	2,157	7,223	77%	4	5,651	305	390	873
Stag1F	10	2,126	10,081	82%	2	8,201	251	220	1,407
Stag2F	12	2,928	13,410	82%	6	10,785	350	376	1,893

Set	Type	SRR	Reads	Ab>100	Non-singles	Singles	Pct.Sgl.
Gingiva	454	SRR041506	4,364	10	119	620	80%
Mouth	454	SRR041522	4,339	5	201	527	72%
Stool	454	SRR041491	4,887	19	212	932	81%
Throat	454	SRR041528	4,444	12	159	653	80%
Tongue	454	SRR041503	5,217	12	177	731	80%
Beer	Illumina	(Bolulich *et al.* set #9).	2.8 M	616	10,955	46,442	81%

Note 3. Software commands, parameters and computational resources.

SN3.1 Mothur command script

Commands were run with mothur v1.27.0, 64-bit under Linux. The flowgram file reads.sff was obtained by the sff-dump.exe utility from the SRA download file. The oligos.txt file contained one line specifying the V5 primer:

```
forward   CCGTCAATTCMTTTRAGT   v35
```

The script was as follows.

```
sffinfo(sff=reads.sff, flow=T)
trim.flows(flow=reads.flow, oligos=oligos.txt, pdiffs=2, bdiffs=1,processors=4)
shhh.flows(file=reads.flow.files, processors=4)
trim.seqs(fasta=reads.v35.shhh.fasta, name=reads.v35.shhh.names, oligos=oligos.txt, pdiffs=2,
bdiffs=1, maxhomop=8, minlength=200, flip=T, processors=4)
unique.seqs(fasta=reads.v35.shhh.trim.fasta, name=reads.v35.shhh.trim.names)
align.seqs(fasta=reads.v35.shhh.trim.unique.fasta, reference=silva.bacteria.fasta, processors=4)
screen.seqs(fasta=reads.v35.shhh.trim.unique.align, name=reads.v35.shhh.trim.unique.names,
group=reads.v35.shhh.groups, end=27659, optimize=start, criteria=95, processors=4)
filter.seqs(fasta=reads.v35.shhh.trim.unique.good.align, vertical=T, trump=., processors=4)
unique.seqs(fasta=reads.v35.shhh.trim.unique.good.filter.fasta,
name=reads.v35.shhh.trim.unique.good.names)
pre.cluster(fasta=reads.v35.shhh.trim.unique.good.filter.unique.fasta,
name=reads.v35.shhh.trim.unique.good.filter.names, group=reads.v35.shhh.good.groups, diffs=2)
chimera.uchime(fasta=reads.v35.shhh.trim.unique.good.filter.unique.precluster.fasta,
name=reads.v35.shhh.trim.unique.good.filter.unique.precluster.names,
group=reads.v35.shhh.good.groups, processors=4)
remove.seqs(accnos=reads.v35.shhh.trim.unique.good.filter.unique.precluster.uchime.accnos,
fasta=reads.v35.shhh.trim.unique.good.filter.unique.precluster.fasta,
name=reads.v35.shhh.trim.unique.good.filter.unique.precluster.names,
group=reads.v35.shhh.good.groups)
system(cp reads.v35.shhh.trim.unique.good.filter.unique.precluster.pick.names final.names)
system(cp reads.v35.shhh.trim.unique.good.filter.unique.precluster.pick.fasta final.fasta)
dist.seqs(fasta=final.fasta, cutoff=0.15, processors=4)
cluster(column=final.dist, name=final.names)
get.oturep(column=final.dist, name=final.names, fasta=final.fasta)
quit()
```

SN3.2 QIIME commands for 454 reads

The QIIME release version was 1.5.0 at the time this work was done (Oct. 2012). This version does not support SRA or FASTQ files that lack barcodes. Following the advice of the

QIIME developers (J. Gregory Caporaso, personal communication), FASTQ files were converted to FASTA using the *convert_fastaqual_fastq.py* script using the new *-c fastq_to_fastaqual* option from developer version 1.5.0-dev (git commit hash 4974f0fe84e6aa9e6f41aa65aacf6ad60bcb6576), downloaded from the QIIME github repository (http://github.com/qiime/qiime). OTUs were built from the FASTA reads using the v1.5.0 *pick_otus.py* script with default parameters.

SN3.3 UPARSE commands

FASTQ quality filtering and the UPARSE algorithm were executed using version 6.1.351_i86linux32 of the usearch binary distribution. Command lines are given in Table S6.

The UPARSE pipeline does not use the previously published algorithms USEARCH, UCLUST or UCHIME.

SN3.4 User-settable parameters for UPARSE.

UPARSE has only three significant user-settable parameters (minimum Q score, trim length and cluster radius) as summarized in Table S6. There are also minor parameters such as gap penalties and the alignment substitution matrix that would rarely if ever be set by a user. Minimum quality (Q_{min}) and truncation length (L) were selected by considering charts similar to Fig SN3.1; see figure caption for discussion.

SN3.5 Computational resources.

All pipelines except AmpliconNoise were executed on a single-CPU, 4-core desktop computer (3GHz Intel Xeon). The UPARSE pipeline required < 1Gb RAM to process all datasets described here, completing in wall-clock times ranging from a few seconds to five minutes. QIIME also required < 1Gb of RAM and completed in times ranging from a few minutes for 454 datasets to a few hours for the Illumina datasets. Mothur required 3Gb of RAM and completed in times ranging from 30 minutes to two hours per 454 dataset. AmpliconNoise required approximately four hours wall-clock time to execute the 454 datasets on a 48-core server-class computer.

Figure SN3.1. Analysis of Even1P 454 reads.

This figure shows an analysis of 454 reads for the Even1P dataset. The average number of expected errors over all reads if truncated at each position was calculated from the Q scores (red line, right-hand y axis). Four different quality filters are considered with different quality score thresholds ($Q_{\min}$). The fraction of reads passing each filter if truncated each position is shown (left-hand y axis). The truncation length L=250 and $Q_{\min}$=16 values (black dot) were selected as a compromise between stringent quality filtering to suppress errors (high Q), keeping as many reads as possible to increase sensitivity to low-abundance sequences (small L and low Q), keeping as many positions as possible to increase phylogenetic discrimination (large L), truncating in order to discard lower-quality regions towards the end of the reads (small L).

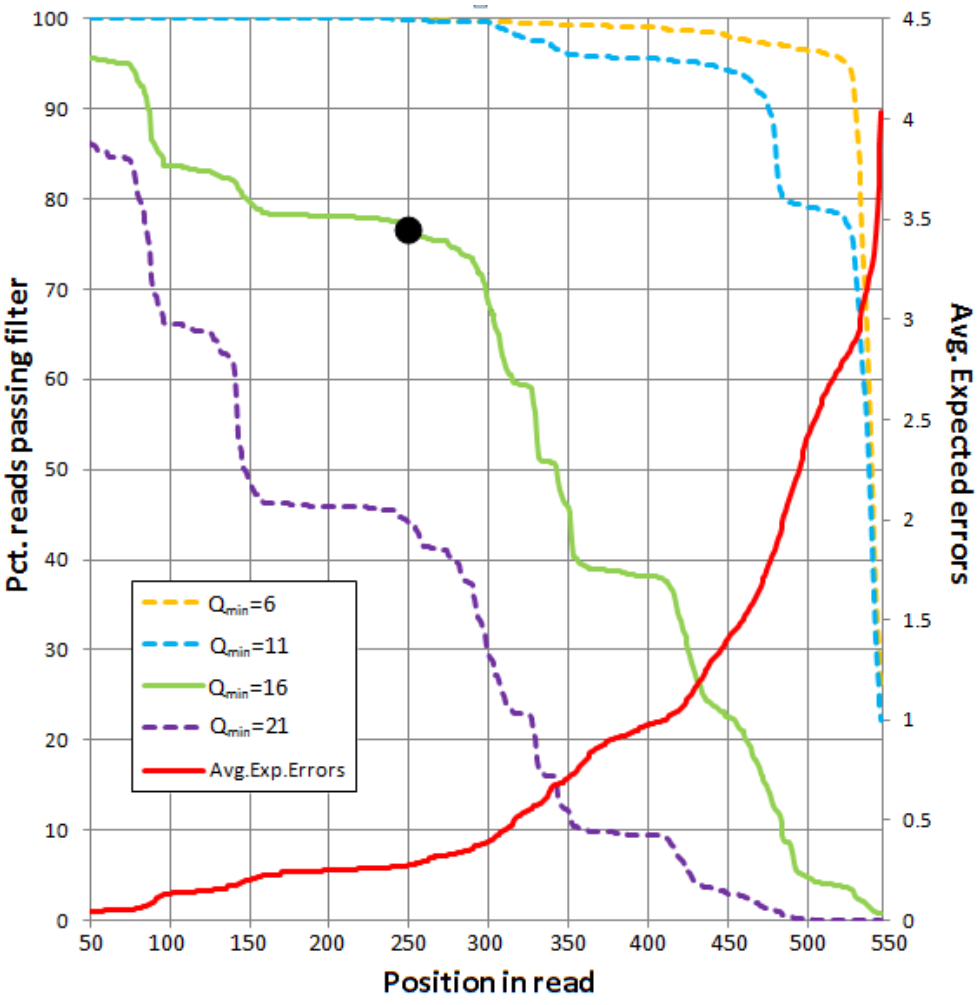


Table SN3.1. UPARSE command lines.

This implementation of UPARSE requires FASTQ-formatted reads from which barcodes have been stripped.

Step	Command line
Merge paired reads.	`usearch -fastq_mergepairs forward.fastq \` `  -reverse reverse.fastq -fastqout merged.fastq`

Merge paired reads allowing no differences in the overlap region.	`usearch -fastq_mergepairs forward.fastq \`
	`  -reverse reverse.fastq -fastqout merged.fastq \`
	`  -fastq_maxdiffs 0`

Quality filtering and length truncation. 200 was used in place of 250 for reverse reads.	`usearch -fastq_filter reads.fastq -fastaout \` ` filtered.fasta -fastq_trunqual 15 -fastq_trunclen 250`
Dereplication.	`usearch -derep_fulllength filtered.fasta \` `   -output derep.fasta -sizeout`
Discard singletons.	`usearch -sortbysize filtered.fasta  \` `  -output derep2.fasta -minsize 2`
UPARSE-OTU	`usearch -cluster_otus derep2.fasta -otus otus.fasta`

Table SN3.2. User-settable parameter values.

This table summarizes the major user-settable parameters in the UPARSE pipeline.

Stage	Parameter	Description
Quality filtering	Q_{min}	Minimum quality score. Default 16.
Length trimming	L	Fixed length for unpaired reads. Values used: 454 L=250, Illumina fwd L=250, rev L=200.
UPARSE-OTU	d_{max}	OTU "radius". Default 3%.

Supplementary References

1. Bokulich, N. A. *et al.* Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. *Nature methods* **10**, 57–9 (2013).

2. Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C. & Knight, R. UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics (Oxford, England)* **27**, 2194–200 (2011).

3. Haas, B., Gevers, D. & Earl, A. Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome research* 494–504 (2011).

4. Minoche, A. E., Dohm, J. C. & Himmelbauer, H. Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems. *Genome biology* **12**, R112 (2011).

5. Gilles, A. *et al.* Accuracy and quality assessment of 454 GS-FLX Titanium pyrosequencing. *BMC genomics* **12**, 245 (2011).

6. Ihrmark, K. *et al.* New primers to amplify the fungal ITS2 region - evaluation by 454-sequencing of artificial and natural communities. *FEMS microbiology ecology* **82**, 666–77 (2012).

7. Edgar, R. C. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* **26**, 2460–1 (2010).

8. Caporaso, J. G. *et al.* QIIME allows analysis of high-throughput community sequencing data. *Nature methods* **7**, 335–6 (2010).

Nature Methods: doi:10.1038/nmeth.2604