DADA2: High-resolution sample inference from Illumina amplicon data

Benjamin J Callahan[1], Paul J McMurdie[2], Michael J Rosen[3], Andrew W Han[2], Amy Jo A Johnson[2] & Susan P Holmes[1]

We present the open-source software package DADA2 for modeling and correcting Illumina-sequenced amplicon errors (https://github.com/benjjneb/dada2). DADA2 infers sample sequences exactly and resolves differences of as little as 1 nucleotide. In several mock communities, DADA2 identified more real variants and output fewer spurious sequences than other methods. We applied DADA2 to vaginal samples from a cohort of pregnant women, revealing a diversity of previously undetected *Lactobacillus crispatus* variants.

The importance of microbial communities to human and environmental health has motivated researchers to develop methods for the efficient characterization of these communities. The most common and cost-effective of these methods is the amplification and sequencing of targeted genetic elements[1]. Amplicon sequencing of taxonomic marker genes such as the 16S rRNA gene in bacteria, the ITS region in fungi, and the 18S rRNA gene in eukaryotes provides a census of a community. Functional diversity can be probed by targeting functional genes[2].

Disentangling biological variation from amplicon sequencing errors presents unique challenges that have prompted the development of amplicon-specific error-correction methods[3–6]. Most of these methods were designed for 454 pyrosequencing and are not applicable to Illumina sequencing.

Errors in Illumina-sequenced amplicon data are currently addressed by quality filtering and the construction of operational taxonomic units (OTUs): clusters of sequences that differ by less than a fixed dissimilarity threshold[7–9] (typically 3%). Lumping together similar sequences reduces the rate at which errors are misinterpreted as biological variation (**Supplementary Fig. 1**), but OTUs underutilize the quality of modern sequencing by precluding the possibility of resolving fine-scale variation[5,10–12]. Fine-scale variation can be informative about ecological niches[10], temporal dynamics[12], and population structure[2]. Fine-scale variation differentiates pathogenic from commensal strains in some cases[13,14] and can contain clinically relevant information for more complex microbiome-associated diseases[15].

We previously introduced the Divisive Amplicon Denoising Algorithm (DADA), a model-based approach for correcting amplicon errors without constructing OTUs[5]. DADA identified fine-scale variation in 454-sequenced amplicon data while outputting few false positives[2,5].

Here we present DADA2, an open-source R package (https://github.com/benjjneb/dada2, **Supplementary Software**) that extends and improves the DADA algorithm. DADA2 implements a new quality-aware model of Illumina amplicon errors. Sample composition is inferred by dividing amplicon reads into partitions consistent with the error model (Online Methods). DADA2 is reference free and applicable to any genetic locus. The DADA2 R package implements the full amplicon workflow: filtering, dereplication, sample inference, chimera identification, and merging of paired-end reads.

We compared DADA2 to four algorithms (Online Methods): UPARSE, an OTU-construction algorithm with the best published false-positive results[9]; MED, an algorithm with the best published fine-scale resolution in Illumina amplicon data[11]; and the popular mothur (average linkage) and QIIME (uclust) OTU methods[7,8].

We benchmarked these algorithms on three mock community data sets: Balanced, HMP, and Extreme (Online Methods and **Supplementary Table 1**), each sequenced at a depth of over 500,000 highly overlapping paired-end Illumina MiSeq 2 × 250 reads. The Balanced community contained 57 bacteria and archaea at nominally equal frequencies[16], the HMP community contained 21 bacteria at nominally equal frequencies[17], and the Extreme community contained 27 bacterial strains at frequencies spanning five orders of magnitude and differing over the sequenced region by as little as 1 nucleotide (nt) (Online Methods and **Supplementary Table 2**). Sequence quality varied, as Balanced demonstrated higher (Mean $Q = 35.9$ forward/33.5 reverse) quality; Extreme, moderate (33.0/29.3) quality; and HMP, lower (32.3/28.7) quality.

We compared output sequences to the known reference sequences present in the reference strains making up these communities. Output sequences that exactly matched a reference sequence were classified as Reference, and those that differed by one mismatch or gap were classified as One Off. Contaminants were identified in the yet unclassified sequences by performing a BLAST search against nucleotides (Online Methods). Sequences with an exact BLAST hit (100% identity, 100% coverage) were classified as Exact, and those with best hits containing one mismatch or gap were classified as One Off. Everything else was classified as Other. We evaluated sensitivity as the proportion of detected reference strains. Note that fine-scale variation was present in all mock communities, as some reference strains contained multiple distinguishable 16S rRNA sequence variants.

We compared the sample sequences output by DADA2 to the OTUs output by UPARSE (**Fig. 1**). Almost all variants with

[1]Department of Statistics, Stanford University, Stanford, California, USA. [2]Second Genome, South San Francisco, California, USA. [3]Department of Applied Physics, Stanford University, Stanford, California, USA. Correspondence should be addressed to B.J.C. (benjamin.j.callahan@gmail.com).

RECEIVED 21 AUGUST 2015; ACCEPTED 13 APRIL 2016; PUBLISHED ONLINE 23 MAY 2016; DOI:10.1038/NMETH.3869

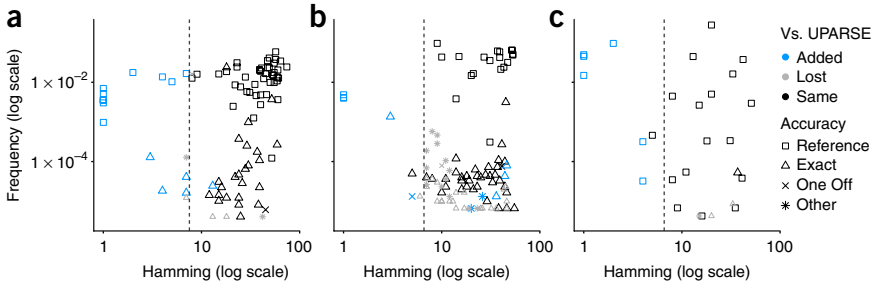

Figure 1 | Comparison of sequence variants inferred by DADA2 with OTUs constructed by UPARSE. (**a**–**c**) The merged sequences output by DADA2 are plotted for three Illumina amplicon data sets: (**a**) Balanced, (**b**) HMP, and (**c**) Extreme. Frequency is plotted on the *y*-axis; Hamming distance to the closest more abundant sequence is plotted on the *x*-axis. Shapes represent accuracy (Online Methods). When variants are well separated from other members of the community, the sequence variants inferred by DADA2 largely coincide with the OTUs output by UPARSE (black). However, DADA2 resolves additional variation (blue), especially within UPARSE's OTU radius (dashed line), while outputting fewer spurious sequences (One Off and Other).

Hamming separation greater than UPARSE's OTU radius (3%) were identified by both algorithms. However, DADA2 identified fine-scale variation that UPARSE did not in both the merged reads (**Fig. 1**) and the forward reads alone (**Supplementary Figs. 2**–**7**). DADA2 accurately resolved sequence variants that differed by a single nucleotide and were present in as few as two reads.

Using merged or forward reads, DADA2 identified more reference sequences and as many or more reference strains than UPARSE in every data set (**Table 1**). DADA2 identified every reference strain in the Balanced and HMP data sets; the Extreme reference strains it missed demonstrate its limits (**Supplementary Note 1**). DADA2 output fewer spurious sequences (Other and One Off) than UPARSE in every data set (**Table 1**).

Minimum entropy decomposition (MED) is a method used to distinguish fine-scale diversity in amplicon data[11]. Like DADA2, MED divides amplicon reads into partitions within which variation is supposed to be artifactual. MED effectively uses a single-site minor-allele frequency threshold to distinguish real variation, and it prevents false positives with a minimum abundance threshold. Thus, while MED identified fine-scale variation, it output more false positives than DADA2 and could not detect rare variants (**Table 1** and **Supplementary Figs. 2**–**7**).

Mothur and QIIME output significantly more spurious sequences than the other methods, although this deficiency was reduced when merging reads (**Table 1** and **Supplementary Figs. 2**–**7**). UPARSE was the most accurate OTU method tested by all measures save the number of reference strains identified in the merged Extreme data set. The spurious output of mothur and QIIME included chimeric and nonchimeric errors (**Supplementary Table 3**).

The residual error rates in the output of DADA2 were very low. For the Balanced data set, DADA2's residual error rate of 2.46×10^{-8} (forward) and 2.53×10^{-8} (merged) compared favorably to the lowest error rates of 5.9×10^{-3} and 5.0×10^{-4} previously reported[16]. For the HMP data set, DADA2's residual error rate of 1.66×10^{-5} (forward) and 2.74×10^{-6} (merged) compared favorably to the previously reported error rates of 9.2×10^{-4} and 4.6×10^{-4} in ref. 18 and 2×10^{-4} (merged) in ref. 17. DADA2 discarded relatively few reads compared with the other methods (**Table 1**).

DADA2's core denoising algorithm was slower than, but comparable to, UPARSE, and DADA2 easily processed Illumina samples on a laptop. For the filtered Balanced forward reads (33,516 unique sequences), UPARSE ran in 9 s, QIIME (uclust) in 17 s, DADA2 in 21 s, mothur in 2 m 26 s, and MED in 2 m 34 s on a 2013 MacBook Pro (Online Methods).

Table 1 | The accuracy of DADA2, UPARSE, MED, mothur, and QIIME on three mock community data sets

				Output sequences					
			Output reads (%)	Total	Reference	Exact	One Off	Other	Reference strains
Balanced	Forward	DADA2	99.2	93	59	33	1	0	57
		UPARSE	99.1	81	48	29	2	2	53
		MED	95.5	86	59	5	22	0	57
		Mothur	96.3	249	44	25	15	165	49
		QIIME	99.2	378	51	34	3	290	54
	Merged	DADA2	96.2	87	57	29	1	0	55
		UPARSE	94.2	76	45	27	2	2	50
		MED	91.1	64	56	6	2	0	54
		Mothur	94.1	108	42	27	11	28	47
		QIIME	94.1	170	45	28	4	93	50
HMP	Forward	DADA2	95.1	151	23	112	8	8	21
		UPARSE	96.7	161	20	123	10	8	21
		MED	80.9	83	23	2	58	0	21
		Mothur	95.4	849	20	177	47	605	21
		QIIME	97.4	1,375	20	177	60	1,118	21
	Merged	DADA2	92.3	67	23	40	2	2	21
		UPARSE	67.7	94	20	59	2	13	21
		MED	64.8	32	23	3	6	0	21
		Mothur	62.1	121	20	82	9	10	21
		QIIME	67.6	290	20	71	8	191	21
Extreme	Forward	DADA2	99.5	68	26	35	3	4	23
		UPARSE	99.5	74	21	40	0	13	21
		MED	86.4	95	16	0	79	0	13
		Mothur	–	–	–	–	–	–	–
		QIIME	99.5	3,237	20	44	73	3,100	20
	Merged	DADA2	97.6	25	24	1	0	0	21
		UPARSE	69.9	23	18	4	0	1	18
		MED	67.6	32	17	0	15	0	14
		Mothur	94.3	44	23	14	0	7	23
		QIIME	69.9	36	19	8	1	8	19

Output sequences were classified as Reference or Exact (true positives) and One Off or Other (false positives) by comparison to the known sequences of these mock communities (reference strains) and comparison to nt to identify contaminants (Online Methods). Mothur failed to complete on the Extreme forward reads.

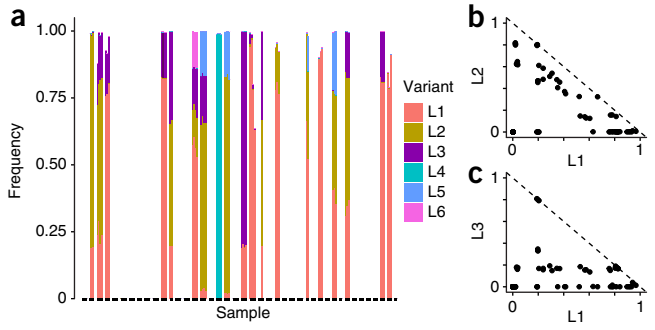

Figure 2 | *L. crispatus* sequence variants in the human vaginal community during pregnancy. DADA2 identified six *L. crispatus* 16S rRNA sequence variants present in multiple samples and a significant fraction of all reads (L1, 19.7%; L2, 11.1%; L3, 6.5%; L4, 3.1%; L5, 1.3%; and L6, 0.4%). (**a**) The frequency of L1–L6 in each sample. Black bars at the bottom link samples from the same subject. (**b**,**c**) The frequency of (**b**) L1 vs. L2 and (**c**) L3 vs. L1, by sample. The dashed line indicates a total frequency of 1.

We further evaluated DADA2 on two longitudinal, Illumina-sequenced data sets: 142 vaginal samples from 42 pregnant women[19] and 360 mouse fecal samples[17]. The vagina is the least diverse human body habitat[1], often dominated by a single *Lactobacillus* OTU[20]. *L. crispatus* is the most common species, and *L. crispatus*-dominated communities have been associated with good health and stability[20]. DADA2 revealed that *L. crispatus* communities are more complex than has generally been recognized: six distinct *L. crispatus* sequence variants were present in substantial abundance in multiple samples we analyzed (**Fig. 2a**). This variation is imperceptible to OTU methods, as these variants differ by just 1 or 2 nt over the sequenced region.

The joint distribution of *L. crispatus* variants strongly suggests that they represent multiple bacterial strains. The distribution of sequence variants was stable over time but differed substantially between women, and distinct ecological relationships appeared to exist between variants. Variants L1 and L2 showed a pattern of mutual exclusion consistent with competition for a common niche (**Fig. 2b**). L1 and L3 showed a pattern more consistent with an absence of direct competition (**Fig. 2c**). The frequency of L1 was independent of the frequency of L3, which strongly tended toward 20%.

The fecal community is more diverse than the vaginal community or mock communities, and so it could present different bioinformatic challenges. When applied to the mouse fecal data set, DADA2 output 389 sequences and UPARSE output 327 OTUs, and 257 of these output sequences were shared. Output sequences were subject to BLAST search against the nt database: 247/257 of the shared sequences, 123/132 of the DADA2-only sequences, and 26/70 of the UPARSE-only sequences were Exact matches. DADA2-only sequences were typically close to other sample sequences: the median Hamming distance of DADA2-only sequences to a more abundant output sequence was 2, whereas for shared sequences it was 15. Sample richness was highly correlated between the methods (Pearson correlation = 0.99), but DADA2 identified more variants (mean ratio = 1.21). These results are consistent with the mock community benchmarking: DADA2 identified more biological variants, especially within UPARSE's OTU radius, while outputting fewer spurious sequences than the UPARSE method.

These comparisons show that DADA2 is more accurate than other methods. DADA2 resolves fine-scale variation better than the method currently considered most robust for that task, but it outputs fewer incorrect sequences than the best OTU method. The precision of DADA2 improves downstream measures of diversity and dissimilarity and could potentially allow amplicon methods to probe strain-level variation.

The output of DADA2 can be clustered into OTUs, but this often eliminates biological information present in the data. OTUs are not species, and their construction is not necessitated by amplicon errors. DADA2 enhances the study of microbial communities by allowing researchers to accurately reconstruct amplicon-sequenced communities at the highest resolution.

METHODS

Methods and any associated references are available in the online version of the paper.

Accession codes. The Extreme data set is available from the SRA under accession number SRX1478507.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We thank M. Schirmer and D. MacIntyre for productive correspondence. This work was supported by the NSF (DMS-1162538 to S.P.H.), the NIH (R01AI112401 to S.P.H.), and the Samarth Foundation (Stanford Microbiome Seed Grant to B.J.C. and S.P.H.).

AUTHOR CONTRIBUTIONS

B.J.C. and S.P.H. designed the research; B.J.C., P.J.M., and M.J.R. implemented the algorithm; B.J.C. performed the analysis; B.J.C., P.J.M., M.J.R., and S.P.H. wrote the paper; and A.W.H. and A.J.A.J. generated the Extreme data set designed by B.J.C., P.J.M., and A.W.H.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

1. Human Microbiome Project Consortium. *Nature* **486**, 207–214 (2012).
2. Rosen, M.J., Davison, M., Bhaya, D. & Fisher, D.S. *Science* **348**, 1019–1023 (2015).
3. Reeder, J. & Knight, R. *Nat. Methods* **7**, 668–669 (2010).
4. Quince, C., Lanzen, A., Davenport, R.J. & Turnbaugh, P.J. *BMC Bioinformatics* **12**, 38 (2011).
5. Rosen, M.J., Callahan, B.J., Fisher, D.S. & Holmes, S.P. *BMC Bioinformatics* **13**, 283 (2012).
6. Bragg, L., Stone, G., Imelfort, M., Hugenholtz, P. & Tyson, G.W. *Nat. Methods* **9**, 425–426 (2012).
7. Schloss, P.D. *et al. Appl. Environ. Microbiol.* **75**, 7537–7541 (2009).
8. Caporaso, J.G. *et al. Nat. Methods* **7**, 335–336 (2010).
9. Edgar, R.C. *Nat. Methods* **10**, 996–998 (2013).
10. Eren, A.M., Borisy, G.G., Huse, S.M. & Welch, J.L.M. *Proc. Natl. Acad. Sci. USA* **111**, E2875–E2884 (2014).
11. Eren, A.M. *et al. ISME J.* **9**, 968–979 (2015).
12. Tikhonov, M., Leach, R.W. & Wingreen, N.S. *ISME J.* **9**, 68–80 (2015).
13. Wang, C., Mitsuya, Y., Gharizadeh, B., Ronaghi, M. & Shafer, R.W. *Genome Res.* **17**, 1195–1201 (2007).
14. McElroy, K., Zagordi, O., Bull, R., Luciani, F. & Beerenwinkel, N. *BMC Genomics* **14**, 501 (2013).
15. Guarner, F. *Nat. Rev. Gastroenterol. Hepatol.* **11**, 647–649 (2014).
16. Schirmer, M. *et al. Nucleic Acids Res.* **43**, e37 (2015).
17. Kozich, J.J., Westcott, S.L., Baxter, N.T., Highlander, S.K. & Schloss, P.D. *Appl. Environ. Microbiol.* **79**, 5112–5120 (2013).
18. Edgar, R.C. & Flyvbjerg, H. *Bioinformatics* **31**, 3476–3482 (2015).
19. MacIntyre, D.A. *et al. Sci. Rep.* **11**, 8988 (2015).
20. Ravel, J. *et al. Proc. Natl. Acad. Sci. USA* **108**, 4680–4687 (2011).

ONLINE METHODS

The Divisive Amplicon Denoising Algorithm. The core denoising algorithm in the DADA2 R package is built on a model of the errors in Illumina-sequenced amplicon reads. This error model quantifies the rate λ_{ji} at which an amplicon read with sequence i is produced from sample sequence j as a function of sequence composition and quality. A Poisson model for the number of repeated observations of the sequence i, parameterized by the rate λ_{ji}, is then used to calculate the p-value of the null hypothesis that the number of amplicon reads (the abundance) of sequence i is consistent with the error model. These p-values are used as the division criteria for an iterative partitioning algorithm, which continues dividing sequencing reads until all partitions are consistent with being produced from their central sequence[5]. We now describe each of these steps in detail.

Sequence comparison. Pairwise sequence alignments are performed by a vectorized implementation of the Needleman-Wunsch algorithm with ends-free gapping. As alignment dominates the computational costs of the algorithm, two heuristics are enabled by default. The first heuristic is the use of a kmer-distance screen before alignment. If the kmer distance between i and j exceeds a user-settable parameter (KDIST_CUTOFF), no alignment is performed[21]. The default value of this parameter was chosen to exclude only pairs of reads with >10% nucleotide mismatch. The second heuristic is banded alignment, which forgoes calculation of potential alignments in which the net number of gaps of one sequence relative to the other exceeds a user-settable parameter (BAND_SIZE). The default value of this parameter was chosen to minimally impact the alignment of sequences with few indels, such as ribosomal RNA genes. Both heuristics can be disabled by the user, and the default values should be re-examined if the algorithm is applied to genetic regions with significantly different characteristics, such as the indel-rich ITS region.

Error model. DADA2 models errors as occurring independently within a read and independently between reads. Under this model, the rate at which an amplicon read with sequence i is produced from sample sequence j is reduced to the product over the transition probabilities between the L aligned nucleotides:

$$\lambda_{ji} = \prod_{l=0}^{L} p(j(l) \rightarrow i(l), q_i(l))$$

The transition probability between aligned nucleotides is allowed to depend on the original nucleotide, substituting nucleotide, and associated quality score, for example, $p(\text{A}\rightarrow\text{C}, 35)$.

After sequence alignment, the error rate λ_{ji} is calculated and stored. If sequences i and j were not aligned because they exceeded the kmer-distance cutoff, λ_{ji} is set to 0.

The abundance *p*-value. The abundance p-value quantifies the notion that sequence i is too abundant to be explained by errors in amplicon sequencing. If sequencing errors are independent across reads, the number of amplicon reads with sequence i that will be produced from sample sequence j is Poisson distributed with expectation equal to an error rate λ_{ji} multiplied by the expected reads of sample sequence j. Let unique sequence i with abundance a_i be in partition j containing n_j reads. Then, conditional on i being read at least once, the abundance p-value is the probability of seeing n_j or more identical reads (ρ_{pois} is the Poisson density function):

$$p_A(j \rightarrow i) = \frac{1}{1-\rho_{\text{pois}}(n_j\lambda_{ji},0)} \sum_{a=a_i}^{\infty} \rho_{\text{pois}}(n_j\lambda_{ji},a)$$

A low p_A indicates that there are more reads of sequence i than can be explained by errors introduced during the amplification and sequencing of n_j copies of sample sequence j.

Note that the abundance p-value is calculated conditional on at least one sequence being observed. As a result, all singleton sequences have an abundance p-value of 1 and are never judged inconsistent with the error model. This means that singletons cannot form their own partitions, and DADA2 will not infer singleton sequences. The effect of this is similar in practice to the UPARSE developer's recommendation to remove singleton sequences before picking OTUs, and in both cases is driven by the difficulty in robustly differentiating singleton errors from real singleton sequences.

The divisive partitioning algorithm. First, amplicon reads with the same sequence are grouped into unique sequences with an associated abundance and consensus quality profile (or dereplicated). The divisive partition algorithm is initialized by placing all unique sequences into a single partition and assigning the most abundant sequence as the center of that partition. All unique sequences are then compared to the center of their partition, error rates are calculated and stored, and the abundance p-value is calculated for each unique sequence. If the smallest p-value, after Bonferroni correction, falls below the user-settable threshold OMEGA_A, a new partition is formed with the unique sequence with the smallest p-value as its center, and all unique sequences are compared to the center of that new partition.

After a new partition is formed, every unique sequence is allowed to join the partition most likely to have produced it (i.e., the partition that produces the highest expected number of that unique sequence). At that point, the division procedure iterates, with each iteration consisting of identifying the unique sequence with the smallest p-value, forming a new partition with that sequence as its center, and reshuffling sequences to their most likely partition.

Division continues until all abundance p-values are greater than OMEGA_A; i.e., all unique sequences are consistent with being produced by amplicon sequencing the center of their partition. The inferred composition of the sample is then the set of central sequences and the corresponding total abundances of those partitions (alternatively, each read is denoised by replacing it with the central sequence of its partition).

A detailed description of the original version of this divisive algorithm is available in Rosen *et al.*[5]. The DADA2 implementation has been slightly simplified for computational speed; notably, there is no longer any construction of 'indel families'.

Error model parameterization. DADA2 depends on a parameterized error model (the 16 × 41 transition probabilities, for example, $p(\text{A}\rightarrow\text{C}, 35)$), but if parameters are not known a priori then DADA2 can estimate them from the data.

doi:10.1038/nmeth.3869

Given an inferred partition of the amplicon sequences, DADA2 records the mismatches between every sequence and the center of its partition and counts each type of mismatch (for example, the number of A→C mismatches where $Q = 35$). The resulting table of observed mismatches represents the errors inferred by DADA2 and can be used to estimate the parameters of the error model. DADA2's default parameter estimation method is to perform a weighted loess fit to the regularized log of the observed mismatch rates as a function of their quality, separately for each transition type (for example, A→C mismatches are fit separately from A→G mismatches). However, the error rate estimation function is a modular part of the algorithm, and users can provide their own R function to estimate the parameters of the error model from the observed mismatches if they prefer a different method.

Alternating estimation until consistency. DADA2's selfConsist mode alternates sample inference (conditional on the parameters of the error model) with parameter estimation (conditional on the inferred sample composition) until convergence, at which point jointly consistent estimates of the error parameters and sample composition are reported. This improves the accuracy of DADA2: while Illumina quality scores are informative, they do not exactly match their textbook definition (**Supplementary Fig. 8**), and we have observed significant variation in this relationship between different runs. The plotErrors function in the DADA2 R package produced **Supplementary Figure 8** and is a useful tool for visualizing the observed and estimated error rates in various data sets.

DADA2 pipeline. The DADA2 R package implements a complete pipeline to turn paired-end fastq files from the sequencer into merged, denoised, chimera-free, inferred sample sequences. Parts of this pipeline can be substituted with outside methods, but there are some structural differences between the DADA2 pipeline and most others. One such difference is that the DADA2 pipeline performs merging of paired-end reads after denoising. This is because the core denoising algorithm uses the empirical relationship between the quality score and the error rates. When reads are merged, this relationship will differ between the forward-only, overlapping, and reverse-only portions of the merged read. That variation interferes with the denoising algorithm, and therefore greater accuracy can be achieved by denoising before merging, albeit at some computational cost.

Filtering. fastqFilter() implements filtering of fastq files that largely recapitulates the usearch fastq_filter command (http://www.drive5.com/usearch/manual/cmd_fastq_filter.html). In short, this function trims sequences to a specified length, removes sequences shorter than that length, and filters based on the number of ambiguous bases, a minimum quality score, and the expected errors in a read[18]. fastqPairedFilter() implements the same trimming and filtering, but applies it to paired reads jointly, only outputting reads where both the forward and reverse reads pass the filter.

Dereplication. derepFastq() imports a fastq file and outputs a dereplicated list of unique sequences and their abundances. derepFastq() also outputs consensus positional quality scores for each unique sequence by taking the average (mean) of the positional qualities of the component reads. These consensus scores are used by the error model of the dada() function.

Denoising. dada() implements the core denoising algorithm described above.

Chimeras. isBimeraDenovo() identifies sequences that are exact bimeras (two-parent chimeras) of more abundant output sequences. Bimeras are identified by performing a Needleman-Wunsch global alignment of each sequence to all more abundant sequences and then searching for combinations of a left-parent and a right-parent that cover the child sequence without any mismatches or internal indels. Child sequences that differ by a single mismatch or indel from the chimeric model are also flagged if the left parent and right parent are both at least 4 nt away from the child sequence.

isBimeraDenovo() is intended to be used after denoising and on exactly inferred sample sequences, rather than on noisy input reads or fuzzy OTUs. It was necessary to implement isBimeraDenovo() because most current stand-alone chimera identification programs are intentionally conservative about identifying chimeras that are relatively close to other more abundant sequences because such sequences are expected to later be joined together in the same OTU (**Supplementary Note 2**). DADA2 does not create OTUs and does differentiate closely related sequence variants; therefore, we implemented this simple, but more sensitive, chimera detection method.

Merging. mergePairs() performs a global ends-free alignment between paired forward and reverse reads and merges them together if they exactly overlap. mergePairs() requires that the input forward and reverse reads read in by derepFastq() are in the same order, a feature which is maintained by fastqPairedFilter(). Note that merging in the DADA2 pipeline happens after denoising, hence the strict requirement of exact overlap since it is expected that nearly all substitution errors have already been removed.

Test data sets. The Balanced community consists of 57 bacteria and archaea from a broad range of habitats. The 16S rRNA gene sequences of most of these strains were well separated (>3%) over the region sequenced. However, the sequences of five strains were identical to those of other more abundant strains, while five strains had a total of seven additional distinguishable sequence variants in their genomes that differed by 1 or 2 nt. There were also two strains that were less than 3% different from more abundant strains. The Balanced data set was downloaded from http://www.ebi.ac.uk/ena/data/view/PRJEB6244, and its construction was described in ref. 16, where it is identified as data set DS 35.

The HMP community consists almost entirely of strains well separated (>3%) over the region sequenced (*Staphylococcus epidermis* and *Staphylococcus aureus* are indistinguishable), most of which colonize the human body. The HMP data set was downloaded from http://www.mothur.org/MiSeqDevelopmentData.html, and its construction was described in ref. 17, where it is identified as the MOCK1 sample in run 130403. This data set was also analyzed in ref. 18.

The Extreme data set was generated for this study. The organisms for the Extreme community include human gastrointestinal tract bacterial isolates (**Supplementary Table 2**). The Extreme data set was intended to include more fine-scale variation than the other mock communities, the members of which were chosen in part for their well-separated 16S rRNA gene sequences. The Extreme strains are all distinguishable over the sequenced

region of the 16S rRNA gene, but some pairs of strains differ by as little as one nucleotide. The Extreme data set is available under SRA accession number SRX1478507.

Extreme strains were grown overnight in liquid broth with the medium recommended from the source culture collection for each respective strain (**Table 1**). An aliquot of the bacterial culture was used to directly amplify the 16S rRNA gene. 1 µl of the bacterial culture was used as template to amplify the V4 region of the 16S rRNA gene using fusion gene primers (515f/806r) that incorporate Illumina adaptor sequences and indexing barcodes[22]. The PCR reaction was carried out in a 25-µL mixture containing 1× HotMaster Mix with 2.5 mM Mg^{2+} (5 PRIME, Gaithersburg, MD), 400 nM forward primer, 400 nM reverse primer, along with the bacterial culture template. The following cycling parameters were used: initial cell lysis and DNA denaturing at 95 °C for 10 min, followed by 30 cycles of 95 °C for 30 s, 50 °C for 30 s, and 72 °C for 30 s, ending with a final annealing step at 72 °C for 10 min. PCR amplicons were cleaned using Agencourt AMPure XP beads (Beckman Coulter, Pasadena, CA) following the manufacturer's instructions. Cleaned PCR amplicons were analyzed and quantified using an Agilent 2100 Bioanalyzer.

Strains were grouped into two taxonomic groups, Firmicutes and Bacteroidetes. Within each taxonomic group, strains were designated for one of six ten-fold dilution groups (**Supplementary Table 2**). PCR amplicons for each strain were first normalized to the same concentration. From there, each amplicon was individually diluted to its respective dilution level, and then all amplicons were pooled. The concentration of the pooled library was quantified using the Quant-iT PicoGreen dsDNA Assay kit (Life Technologies, Carlsbad, CA) and analyzed on an Agilent 2100 Bioanalyzer. The pooled library was diluted to 4 nM, and then Illumina's protocol for preparing libraries for sequencing on the MiSeq was followed. The final concentration of the library was diluted to 6 pM with ~20% PhiX spiked in to account for the low base-diversity library. The final pooled library was sequenced on an Illumina MiSeq with a MiSeq Sequencing Reagent Kit v3 to obtain 250-bp paired-end reads using custom sequencing primers as described in ref. 22.

Workflow on test data. A common filtering and trimming was performed before applying each method: The DADA2 fastqPairedFilter (paired reads) and fastqFilter (forward reads only) functions were used to remove sequences with Ns or more than two expected errors[18], and to trim the first 20 nt and the last 10 nt (forward reads) or 10–50 nt (reverse reads) depending on the quality profile of the data.

The usearch command fastq_mergepairs with a minimum overlap of 20 bases and maximum differences of 1 was used to merge the filtered forward and reverse reads for further analysis by UPARSE, MED, and QIIME. Mothur used its native read merging function make.contigs. DADA2 denoised the forward and reverse reads independently and then merged them with its mergePairs function.

Chimeras were removed from the denoised output of DADA2 and MED by isBimeraDenovo in the DADA2 R package, as this tool is intended for the exactly inferred sequences output by these methods. UPARSE has built-in chimera removal. The uchime method included in mothur and QIIME was used to remove chimeras for those pipelines[23].

A list of output sequences and associated abundances was obtained for each algorithm. For DADA2 these were the inferred sample sequences; for UPARSE, mothur, and QIIME, the representative OTU sequences; and for MED, the representative sequences of its 'nodes'.

We also removed singleton OTUs from the outputs of mothur and QIIME. The DADA2, UPARSE, and MED pipelines all decline to call singleton variants, so removing singletons from mothur and QIIME allows a cleaner comparison between methods. Additionally, nearly all of the singleton variants output by mothur and QIIME were spurious, so removing singletons improved their reported accuracy.

Software versions. DADA2 version 0.99.8, usearch version 8.1.1831 (implements UPARSE), MED version 2.0, mothur version 1.36.1, and QIIME version 1.9.1.

Specificity. Output sequences were first compared to the known 16S rRNA gene reference sequences of the members of each mock community. If an output sequence matched a reference sequence, it was classified as Reference; and if it had one mismatch or gap to a reference sequence, it was classified as One Off. Output sequences that were at least Hamming distance 2 from any reference sequence were then BLASTed against the nr/nt database. If the best hit was an exact match covering the full output sequence, it was classified Exact. If there was a single mismatch or indel, it was classified One Off. Output sequences that remained unclassified to this point were classified Other.

We included the BLAST against nr/nt step because even amplicon sequencing data from communities with a putatively known reference composition will contain contaminant sequences. Contaminants are real, albeit unwanted, biological variation and should be identified when correcting amplicon errors. While the nr/nt database is imperfect, it is reasonable to expect that Exact matches are far more likely to be real variants than are Others. Output sequences classified as Other and output sequences classified as One Off that differed by one substitution from a more abundant output sequence were considered a proxy for false positives. Output sequences classified as Reference or Exact were considered true positives.

Sensitivity. We compiled the 16S rRNA gene sequences (reference sequences) for the intended members of each mock community (reference strains). The presence of each reference strain was confirmed by checking that at least one read matching one of its 16S rRNA gene sequences was present in the filtered data set. If no such read existed, that strain was removed from the reference list.

Output sequences were compared to the list of reference sequences. If any output sequence matched any 16S rRNA gene sequence present in a strain, that reference strain was considered to have been identified.

Time benchmarking. When benchmarking computational time, we attempted to isolate the core sample inference algorithms for each pipeline as much as possible. Thus, for the time benchmarking we applied each algorithm to an identically prepared set of sequences: the filtered forward-only reads from the Balanced data set, including singletons. We did not include chimera identification in this benchmarking. The specific commands benchmarked are listed at the end of the Balanced workflow.

doi:10.1038/nmeth.3869

Note that preprocessing steps, such as discarding singletons and reference-based chimera removal, can substantially reduce subsequent computation time for all of these methods.

Analysis of vaginal and fecal samples. The 16S rRNA gene amplicon data from human vaginal samples in ref. 19 (2.13 M paired-end Illumina Miseq reads in 157 samples) and from mouse feces in ref. 17 (3.65 M paired-end Illumina Miseq reads in 362 samples) were analyzed with the DADA2 pipeline outlined above. First, the demultiplexed fastq files were filtered and trimmed in the same manner as the test data sets. Each sample was dereplicated, a portion of the data set was used to estimate the error parameters, and dada() was applied to the full pooled data set using those inferred error parameters. isBimeraDenovo() was used to remove chimeras.

For the human vaginal samples, output sequences that appeared in at least two samples and at least 0.3% of the total reads were taxonomically identified by BLAST. Further analysis focused on the six *L. crispatus* sequence variants identified by this procedure.

Code availability. The DADA2 R package is open source and available on github (https://github.com/benjjneb/dada2). Package binaries are available through Bioconductor as of its 3.3 release. R markdown files implementing the benchmarking on the Balanced, HMP, and Extreme data sets, as well as the analyses of the human vaginal and mouse fecal samples, are available from the Stanford Digital Repository, https://purl.stanford.edu/mh194vj6733.

21. Sun, Y. *et al. Nucleic Acids Res.* **37**, e76 (2009).
22. Caporaso, J.G. *et al. ISME J.* **6**, 1621–1624 (2012).
23. Edgar, R.C., Haas, B.J., Clemente, J.C., Quince, C. & Knight, R. *Bioinformatics* **27**, 2194–2200 (2011).