Error filtering, pair assembly and error correction for next-generation sequencing reads

Edgar, Robert C.; Flyvbjerg, Henrik

Published in:
Bioinformatics

Link to article, DOI:
10.1093/bioinformatics/btv401

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Bioinformatics, 2015, 1–7
doi: 10.1093/bioinformatics/btv401
Advance Access Publication Date: 2 July 2015
Original Paper

OXFORD

Sequence analysis

Error filtering, pair assembly and error correction for next-generation sequencing reads

Robert C. Edgar[1,*] and Henrik Flyvbjerg[2]

[1]Tiburon, CA 94920, USA and [2]Department of Micro- and Nanotechnology, Technical University of Denmark, DK-2800 Lyngby, Denmark

*To whom correspondence should be addressed.

Associate Editor: Inanc Birol

Received on March 26, 2015; revised on May 28, 2015; accepted on June 27, 2015

Abstract

Motivation: Next-generation sequencing produces vast amounts of data with errors that are difficult to distinguish from true biological variation when coverage is low.
Results: We demonstrate large reductions in error frequencies, especially for high-error-rate reads, by three independent means: (i) filtering reads according to their expected number of errors, (ii) assembling overlapping read pairs and (iii) for amplicon reads, by exploiting unique sequence abundances to perform error correction. We also show that most published paired read assemblers calculate incorrect posterior quality scores.
Availability and implementation: These methods are implemented in the USEARCH package. Binaries are freely available at http://drive5.com/usearch.
Contact: robert@drive5.com
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Next-generation sequencing (NGS) machines produce reads of lengths tens to thousands of bases. In an NGS read, the base call at position i is assigned an estimated error probability p_i represented as an integer-rounded Phred (quality) score $Q_i = -10 \log_{10} p_i$. Typical base call error rates for current machines range from 0.1 to 10% (Glenn, 2011). When coverage is high, error detection and correction can be achieved by aligning reads to each other (*de novo* assembly) or to a closely related well-known sequence (reference-based assembly). When coverage is low or a complete set of reference sequences is not available, sequencing error is difficult to distinguish from true biological variation. Errors can be reduced by quality filtering, i.e. by discarding or truncating reads with low-quality base calls, by merging overlapping paired-end reads and, in the case of amplicon reads, by clustering.

Quality filtering is often used in data analysis of next-generation reads but is rarely regarded as a method in its own right which should be designed and validated as a separate step. Published methods for quality filtering typically use *ad-hoc* criteria such as imposing a maximum on the number of bases with less than a given Q score (Bokulich *et al.*, 2013). That article focuses on filtering but does not attempt to measure error rates. Other articles describing analysis pipelines for amplicon reads (e.g. Kozich *et al.*, 2013; Schloss *et al.*, 2011) report the mean error rate but do not consider the tail of the error distribution, which we believe to be important due to the spurious clusters and consequent inflated estimates of diversity (spurious 'rare biosphere') obtained when the tail is not adequately controlled (Huse *et al.*, 2010).

When paired-end reads overlap, an improved prediction of the sequence in the overlapping region can be obtained by aligning the forward and reverse read. Previously published paired-read mergers include SHERA (Rodrigue *et al.*, 2010), FLASH (Magoč and Salzberg, 2011), PANDAseq (Masella *et al.*, 2012), COPE (Liu *et al.*, 2012) and PEAR (Zhang *et al.*, 2014). Of these, only PANDAseq included quality filtering in addition to merging.

Error correction methods for amplicon pyrosequencing reads include PyroNoise (Quince *et al.*, 2009), AmpliconNoise (Quince *et al.*, 2011) and an unnamed method (Reeder and Knight, 2010) that uses a greedy algorithm based on an abundance sort. These methods all require flowgrams and therefore cannot be applied to other technologies such as Illumina. Single-linkage pre-clustering (Huse *et al.*, 2010) and the pre-clustering method of (Kozich *et al.*,

2013) also use abundance differences between closely related sequences to correct errors.

In this work, we use the expected number of errors in a read as a measure of quality for filtering and show that this method is effective at reducing error rates, especially in the tail of the distribution. We show how to calculate the posterior quality scores when read pairs are merged and demonstrate that most previous merging methods calculate incorrect scores. We also describe and validate UNOISE, a new amplicon error-correction method which can be applied to any type of NGS read.

2 Methods

2.1 Quality filtering

Filtering is commonly used in amplicon sequencing applications such as marker gene metagenomics (Edgar, 2013) and immune system repertoire analysis (Jiang *et al.*, 2013) where it is generally not possible to identify reads derived from the same template sequence, ruling out error detection and correction by assembling contigs as typically done in genome sequencing.

2.2 Expected errors

For a given read, the expected number of errors (E) is defined to be the mean number of errors that would be observed in a very large collection of sequences where the error rate at each position is given by its quality score, assuming that errors at different positions occur independently. We prove below that E is the sum of error probabilities:

$$E = \sum_i p_i = \sum_i 10^{-Q_i/10}. \tag{1}$$

E is a real rather than integer value and may be less than 1. We will also show that the most probable number of errors is well approximated by floor(E), i.e. the largest integer $\leq E$.

A natural approach to quality filtering is to impose a maximum value $E_{\max}$ on the expected number of errors so that reads with high E (low quality) are discarded. Motivated by the goal of suppressing reads with larger numbers of errors, we could also consider setting a threshold on the probability $P^+(k|p_1 \dots p_L)$ that a read of length L has at least k errors. For example, 97% is a commonly used identity threshold in marker gene sequencing (Huse *et al.*, 2010), and we might therefore wish to discard reads which have a relatively high probability of having $\geq$3% errors. It turns out (proof below) that a P^+ filter is equivalent to an $E_{\max}$ filter to a good approximation.

The threshold $E_{\max} = 1$ is a natural choice as the most probable number of errors is zero when $E < 1$, and we therefore used $E_{\max} = 1$ for the tests reported here.

2.3 Proofs

For an informal, intuitive proof that E is the sum of error probabilities, consider a large collection of M reads, all having the same set of Q scores. Let K_i be the total number of errors found at position i. Then by definition $p_i = K_i/M$ and hence $K_i = M\,p_i$. The total number of errors K in all positions is:

$$K = \Sigma_i K_i = \Sigma_i M p_i$$

and in the limit of very large M the expected number of errors (E) is, by definition, the mean over the collection:

$$E = K/M = (1/M)\sum_i (MP_i) = \sum_i p_i. \tag{2}$$

For a more formal analysis, we obtain the probability of exactly k errors by summing over all possible ways of distributing k errors into L positions, i.e. over all combinations of k positions selected from L. The probability of a given combination occurring is the product of the probabilities for each position, i.e. p_i for an error or $(1 - p_i)$ for a correct call. This gives the so-called *Poisson binomial distribution* (Wang, 1993):

$$B_{\text{Pois}}(k; p_{1,\dots}\, p_L) = \sum_{k_1=0}^{1}\sum_{k_2=0}^{1}\cdots\sum_{k_L=0}^{1}\delta_{k,\sum_{i=1}^{L}k_i}\prod_{j=1}^{L}p_j^{k_j}(1-p_j)^{1-k_j}. \tag{3}$$

Here, k_i is zero or one, giving the number of errors at position i.

The Kronecker delta

$$\delta_{k,\sum_{i=1}^{L}k_i}$$

is zero unless k equals the sum of k_i, in which case it is one.

By summing over the possible number of errors, which ranges from 0 to L, and rearranging the sums in Equation (3), we again find that the expected number of errors is the sum of error probabilities:

$$\begin{aligned}\langle k\rangle &= \sum_{k=0}^{L} k B_{\text{Pois}}(k; p_1\ \dots\ p_L)\\ &= \sum_{k_1=0}^{1}\sum_{k_2=0}^{1}\cdots\sum_{k_L=0}^{1}\left(\sum_{i=1}^{L}k_i\right)\prod_{j=1}^{L}p_j^{k_j}(1-p_j)^{1-k_j}\\ &= \sum_{i=1}^{L}\sum_{k_i=0}^{1}k_i p_i^{k_i}(1-p_i)^{1-k_i}\\ &= \sum_{i=1}^{L} p_i.\end{aligned} \tag{4}$$

We consider the regime $E \ll L$ to cover all cases of practical interest. A theorem due to Le Cam (1960) states that in this limit the Poisson binomial distribution is well approximated by the simpler and more familiar Poisson distribution with parameter E:

$$B_{\text{Pois}}(k; p_1, \dots\ p_L) \to \text{Pois}(k; E) = e^{-E}\frac{E^k}{k!} \quad \text{for} \quad \sum_{i=1}^{L} p_i^2 \to 0. \tag{5}$$

The Poisson distribution satisfies the following relation:

$$\frac{\text{Pois}(k; E)}{\text{Pois}(k-1; E)} = \frac{E}{k}. \tag{6}$$

It follows from Equation (6) that as long as $k < E$, Pois(k; E) increases with k, and it decreases with k for $k > E$. Thus, Pois(k; E) has its maximal value as function of k at floor(E) (i.e. the largest integer $\leq E$), and floor(E) is therefore the most probable number of errors when $E \ll L$. It follows from Equation (5) that when $E \ll L$, the probability $B_{\text{Poiss}}(k)$ that k errors occur is a function only of E and k, and $P^+(k) = \sum_{j \geq k} P(j)$ is therefore also a function of E and k. A filter which applies a threshold to $P^+(k)$ is therefore equivalent to an $E_{\max}$ filter. The equivalent value of $E_{\max}$ can be calculated from k and the threshold value of P^+ using Equation (5).

2.4 Posterior error probability for a merged base call

The most probable correct base call and posterior error probability for an aligned pair of bases is calculated as follows. Define p_x and p_y

to be the error probabilities and X and Y to be the letters in the forward and reverse reads, respectively. The five possible outcomes are: both base calls correct (**C**), forward correct and reverse wrong (**F**), forward wrong and reverse correct (**R**), both wrong and disagree (**W**) and both wrong and agree (**G**). The prior probabilities of these outcomes are as follows:

$$\begin{aligned} P(\mathbf{C}) &= (1-p_x)(1-p_y), \\ P(\mathbf{F}) &= (1-p_x)p_y, \\ P(\mathbf{R}) &= p_x(1-p_y), \\ P(\mathbf{W}) &= 2p_xp_y/3, \\ P(\mathbf{G}) &= p_xp_y/3. \end{aligned} \tag{7}$$

(For proofs see Supplementary Equations S8–S12 and Fig. S1.) We observe that the base calls agree (**A**, i.e. a match) or disagree (**D**, i.e. a mismatch). Agreement is **C** or **G**, disagreement is **F**, **R** or **W**. Note that $P(\mathbf{A}) = P(\mathbf{C}) + P(\mathbf{G})$, $P(\mathbf{D}) = P(\mathbf{F}) + P(\mathbf{R}) + P(\mathbf{W})$, $P(\mathbf{A}|\mathbf{C}) = 1$, $P(\mathbf{A}|\mathbf{G}) = 1$. By Bayes' theorem,

$$\begin{aligned} P(\mathbf{G}|\mathbf{A}) &= P(\mathbf{A}|\mathbf{G})P(\mathbf{G})/P(\mathbf{A}) \\ &= P(\mathbf{G})/P(\mathbf{A}) = P(\mathbf{G})/\Big(P(\mathbf{C}) + P(\mathbf{G})\Big) \\ &= (p_xp_y/3)/(1-p_x-p_y+4p_xp_y/3). \end{aligned} \tag{8}$$

(Proof: see Supplementary Equation S21 and Fig. S2). $P(\mathbf{G}|\mathbf{A})$ is the probability that X is incorrect given agreement between the two base calls, i.e. the posterior error probability for a matched position in the alignment.

For the case where we observe a disagreement (mismatch), we use the convention that $p_x < p_y$, so that the merged base call is X. By similar reasoning, the mismatch error probability is

$$\begin{aligned} P(\mathbf{R}|\mathbf{D}) + P(\mathbf{W}|\mathbf{D}) &= [P(\mathbf{R}) + P(\mathbf{W})]/P(\mathbf{D}) \\ &= [P(\mathbf{R}) + P(\mathbf{W})]/[P(\mathbf{F}) + P(\mathbf{R}) + P(\mathbf{W})] \\ &= p_x(1-p_y/3)/(p_x + p_y - 4p_xp_y/3). \end{aligned} \tag{9}$$

(Proof: see Supplementary Equation S26 and Fig. S3).

2.5 Incorrect PANDAseq posterior calculation

Let X and Y be calls of unrelated bases with true bases X^* and Y^* and error probabilities p_x and p_y, respectively. Suppose the predicted bases are the same in the two calls, i.e. $X = Y$, and the true bases are also identical ($X^* = Y^*$). This happens if (i) both base calls are correct or (ii) both calls are incorrect and the true letters are identical, hence

$$P(X^* = Y^* | X = Y) = (1-p_x)(1-p_y) + p_xp_y/3. \tag{10}$$

(Proof: see Supplementary Equation S28 and Fig. S5).

Similarly, if the true bases are the same but the predicted bases are different, the cases are (i) X is correct and Y is wrong while the true base is the same or (ii) Y is correct and X is wrong while the true base is the same or (iii) both calls are wrong but the true bases are the same, so

$$\begin{aligned} &P(X^* = Y^* | X \neq Y) \\ &= (1-p_x)p_y/3 + (1-p_x)p_y/3 + 2p_xp_y/9. \end{aligned} \tag{11}$$

(Proof: see Supplementary Equation S29 and Fig. S7).

PANDAseq calculates posterior Q scores using error probability $p = 1 - P(X^* = Y^*)$ using Equation (10) for a match and Equation (11) for a mismatch. This is incorrect. $P(X^* = Y^* | X = Y)$ is not the probability that the base calls are correct if a match is observed; rather, it is the probability that the true bases are the same, which includes the error case $X^* = Y^*$ *and* $X = Y$ *and* $X \neq X^*$. Also, those equations are derived assuming that the bases are independent of each other, but in fact X and Y in the overlapping segment of a paired read are observations of the same base. Therefore, $P(X^* = Y^*) = 1$ when merging, regardless of whether a match or mismatch is observed. The typical effect of the calculation made by PANDAseq is to reduce Q scores at positions where the base calls in the forward and reverse reads match, when in fact scores should increase to reflect that two independent observations of the same base agree on the prediction (Fig. 1, Supplementary Equation S22). Thus, the quality of a high-quality pair decreases after merging by PANDAseq when in fact it should increase. This will degrade the effectiveness of quality filters and SNP callers that assume quality scores are predictive of error frequencies.

2.6 UNOISE algorithm for amplicon error correction

When amplified libraries are sequenced, reads derived from a given biological template can be visualized as an 'error cloud' surrounding the correct sequence. An amplicon may contain polymerase chain reaction (PCR) errors including point errors and chimeras (Haas *et al.*, 2011), producing a 'daughter cloud' connected to the correct biological sequence (Fig. 2). Thus, if a sequence B has a small number of differences (d) compared with a more abundant sequence R, this may be because B has d errors and is derived from the same amplicon as R, and R is probably the correct sequence of that amplicon. This observation has previously been exploited to attempt error correction. For example, the preclustering step of (Kozich *et al.*, 2013) uses $d = 1\%$. However, it is the small minority of outliers in the tail of the error distribution that are responsible for inducing spurious operational taxonomic units (OTUs) obtained by clustering the reads and thus inflating estimates of diversity (Edgar, 2013). These outliers (call them 'harmful') will not be corrected if d is less than the OTU threshold. If errors occur at random, then almost all of the harmful outliers will be singletons, and discarding singletons have been shown to be an effective filter (Edgar, 2013).

Here, we introduce a new denoising algorithm (UNOISE) which allows larger d if the abundance skew, defined as $w = \text{abundance}(R)/\text{abundance}(B)$ (Edgar *et al.*, 2011), is sufficiently large. UNOISE was implemented as a variant of UCLUST (Edgar, 2010) as follows. A database of putatively correct sequences (D) is initially empty. Unique read sequences are compared with D of decreasing abundance. If a read (B) matches a database sequence (R), i.e. if $d \leq d_{\max}$ and $w \geq w_{\min}$, then the abundance of R is

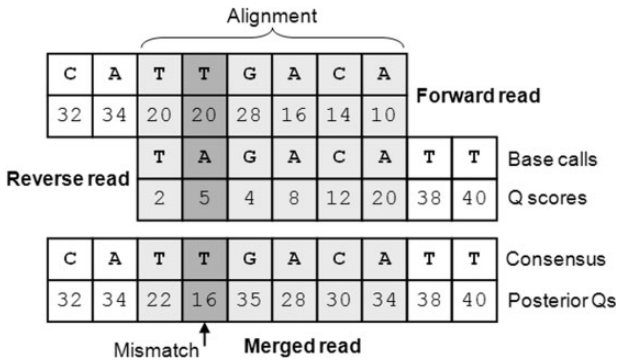

Fig. 1. Paired read merging. The forward read is aligned to the reverse-complemented reverse read. If both reads agree on a base call, the Q score increases due to the increased confidence in the base call per Equation (8). If there is a mismatch, the base call with higher Q is chosen and the posterior Q score is reduced according to Equation (9)

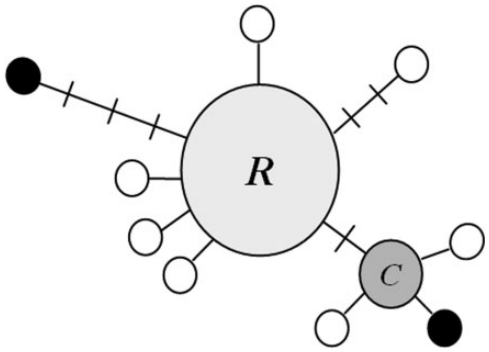

Fig. 2. Conceptual structure of the 'error cloud' due to a given biological template sequence *R*. Circles represent unique read sequences, the size of a circle indicates its abundance (i.e. frequency in the reads). Errors are due to amplification (polymerase chain reaction [PCR] point errors and chimeras) and sequencing. With typical error rates, most incorrect sequences are singletons with a single difference; a few have more differences (indicated by tick marks). *C* is an amplicon containing a PCR error, e.g. a chimera. Amplicons with PCR errors have 'daughter' clouds due to sequencing error. In the tail of the distribution, reads are >3% diverged from *R* (black circles)

increased by that of B, and B is discarded; otherwise, B is added to D. For this work, we chose $d_{\max} = 5$ and $w_{\min} = 10$. These were our first guess at intuitively reasonable and conservative values; we did not try other values because we do not believe valid parameter tuning is possible on the available data. To tune parameters, we need data for which correct sequences are known, i.e. reads of clones or a mock community. However, we expect the biological sequences in such data to be less diverse and more widely separated in sequence space (i.e. to have more differences with their closest neighbors) compared with samples collected *in vivo*. On mock data, an optimal value of $d_{\max}$ will tend to be large because this will include more sequences with larger numbers of errors in the inferred cloud without including the nearest true sequence. In contrast, in reads of *in vivo* samples, we expect nearest neighbors with correct sequences to be closer. Thus, larger values of $d_{\max}$ would increase the number of false positives because neighbors that are correct biological sequences with low abundance would be misidentified as having errors. By the same reasoning, we expect tuning of $w_{\min}$ on mock data to favor small values, which would again increase the number of false positives with *in vivo* data.

3 Results

3.1 Test data

For testing, we used three sets of overlapping paired reads and named them MOCK1, MOCK2 and PHIX, respectively. MOCK1 contains amplicon reads of the V4–V5 region of the 16S rRNA gene in the artificial ('mock') community samples in run 130403 of Kozich *et al.* (2013). MOCK2 contains amplicon reads of the V4 region of the 16S gene from (Bokulich *et al.*, 2013). The MOCK1 and MOCK2 reads were filtered by UCHIME (Edgar *et al.*, 2011) to suppress reads of chimeric amplicons, which require specialized detection algorithms. PHIX is a subset of reads containing PhiX bacteriophage sequences in the data of Reveillaud *et al.* (2014). Reads of the PhiX spike-in library were identified as hits to the PhiX genome sequence (GenBank accession NC_001422.1) with at least 90% identity covering at least 90% of the read. These reads have no bases with $Q < 12$, suggesting that quality filtering using this threshold was performed prior to posting. We would have preferred to use unfiltered reads but were unable to identify a published, unfiltered dataset containing PhiX spike-in reads. MOCK1 and MOCK2 appear to contain unfiltered reads.

3.2 Tested methods

We implemented our own methods in USEARCH (Edgar, 2010) and compared quality filtering and merging with two popular software packages, QIIME (Caporaso *et al.*, 2010) and PANDAseq. We also validated the posterior Q score calculations of COPE, FLASH, PEAR and SHERA. At the time this work was performed, QIIME did not support merging and we therefore tested its filtering script on paired reads after merging by USEARCH. See Supplementary Material for software versions and command lines. We also tested using the forward reads alone, simulating an experiment where paired reads are not available or do not overlap.

3.3 Measurement of error rates

We measured errors by aligning reads (or merged read sequences) to the appropriate reference sequences, i.e. the PhiX genome and the known 16S sequences in the mock communities. We used global alignments to ensure that all bases in the read were included and considered mismatches to be errors. We calculated the mean error rate after filtering as the number of mismatches divided by the number of bases. We also considered the tail of the distribution, which is especially important in marker gene sequencing experiments where reads with high error rates induce spurious clusters and inflate estimates of diversity, even if present only in very low abundance (Edgar, 2013). We considered >3% errors to be the tail as a 3% clustering threshold is conventionally used in marker gene sequencing (Edgar, 2010; Kozich *et al.*, 2013) in which case a read with >3% errors is certain to induce a spurious cluster, noting that reads with fewer errors may also do this. Results are summarized in Tables 1 and 2.

3.4 Correlation of measured and expected errors

Figure 3a and b shows that E is predictive of the number of measured errors in the MOCK1 forward and reverse reads, respectively. These results show that E tends to be an underestimate on MOCK1; for example, the median number of measured errors for reads with $4.5 \leq E < 5.5$ is 18. We obtained similar results on the PHIX reads, and with MOCK2, we observed a correlation where E tends to overestimate rather than underestimate the number of errors (Supplementary Fig. S10). This reflects that the accuracy and biases of Q scores may vary between sequencing runs.

3.5 Error rates after merging and filtering

Table 1 summarizes results obtained by QIIME, PANDAseq and USEARCH on the test datasets, showing that merging by USEARCH followed by filtering with $E_{\max} = 1$ achieves a dramatic reduction in the number of reads with >3% errors compared with QIIME and PANDAseq. UNOISE achieves a further substantial improvement in the number of reads with zero errors in the case of amplicon reads. Figure 3b and c shows the head (<3% errors) and tail (>3% errors) of the measured error distribution after paired-read merging and filtering of the MOCK1 reads. We obtained similar results for MOCK2 and PHIX (Supplementary Fig. S2).

3.6 Posterior *Q* scores

With the exception of PANDAseq, the equations used to calculate posterior Q scores by the tested read mergers are not specified in their publications. We therefore reverse engineered the calculation by generating sets of simulated read pairs of length 150 nt with

Table 1. Summary of results on the test datasets

Dataset	Method	Err. rate (%)	Reads	>3% errs.
MOCK1 (Fwd)	Raw reads	1.2	737 660	79 481
	QIIME/F	0.99	737 134	70 319
	USEARCH/F	0.22	392 917	2687
	UNOISE	0.092	611 176	1042
MOCK1	Merged	2.4	737 660	196 035
	QIIME/MF	2.0	737 102	165 642
	PANDAseq	1.9	717 064	157 797
	USEARCH/MF	0.23	186 695	562
	UNOISE	0.046	354 220	491
MOCK2 (Fwd)	Raw reads	0.54	7 420 628	237 274
	QIIME/F	0.33	7 033 106	73 228
	USEARCH/F	0.19	2 912 861	401
	UNOISE	0.009	6 565 970	251
MOCK2	Merged	0.35	7 420 628	144 584
	PANDAseq	0.34	7 393 489	138 297
	QIIME/MF	0.33	7 352 181	126 909
	USEARCH/MF	0.19	6 736 514	1609
	UNOISE	0.016	7 017 279	1226
PHIX (Fwd)	Raw reads	0.25	1 201 502	8179
	QIIME/F	0.68	1 201 435	8179
	USEARCH/MF	0.20	1 044 123	3945
PHIX	Merged	0.22	1 094 091	4681
	QIIME/MF	0.22	1 094 091	4681
	PANDAseq	0.22	1 139 895	4375
	USEARCH/MF	0.17	983 832	550

Fwd indicates forward reads only. *Method* is one of raw reads (no processing), UNOISE, QIIME/F (QIIME quality filtering only), merged (merge by USEARCH only, no filtering), USEARCH/F (USEARCH filtering by $E_{max} = 1$), USEARCH/MF (USEARCH merge and filtering by $E_{max} = 1$), PANDAseq (merging and filtering with default parameters) and QIIME/MF (QIIME quality filtering of reads merged by USEARCH). *Reads* is the number of reads after any merging and/or filtering. *Err. rate* is the number of measured errors divided by the total number of bases and >3% errs. gives the number of reads with at least 3% measured errors. UNOISE assumes amplicon data, so is not applicable to PHIX.

50 bp overlaps having exactly one mismatch. The sets were generated such that all pair-wise combinations of Q occur with both matches and mismatches. The merged reads from each program were used to construct tables giving the posterior Q for all possible combinations of forward Q, reverse Q and agreement (match or mismatch). Of the tested programs, only SHERA correctly calculated the posterior Q scores in the aligned region. It was readily apparent from the tables that FLASH and PEAR use simple heuristics: FLASH takes the maximum Q score at a position with a match and the difference Q_{max}–Q_{min} at a position with a mismatch, while PEAR uses the sum of the Q scores at a position with a match and the minimum at a position with a mismatch. We were unable to deduce the rules used by COPE, noting that source code is not provided, contrary to the paper's statement that the software is open source. However, it is clear from the reverse-engineered tables that the posterior Q scores generated by COPE are sometimes badly wrong (Supplementary Table S1).

3.7 PANDAseq false-positive merges

We noticed that PANDAseq sometimes merged read pairs that did not overlap, i.e. generated false-positives merges. To investigate this systematically, we generated simulated read pairs containing unrelated random sequences with all Q scores set to 20 ($p^{correct} = 0.99$).

Table 2. Results of merging, filtering, discarding singletons and denoising on the MOCK2 dataset

Pairs?	Filter	Ab.	Den.	Reads	Uniques	OTUs
Fwd only	(No)	1	n	7 420 628	1 414 383	378 537
			Y	7 420 628	439 597	52 452
		2	n	6 158 319	152 086	203
			Y	6 158 319	5055	66
	$E_{max} = 1$	1	n	2 912 861	147 539	110
			Y	2 912 861	141	35
		2	n	2 803 435	38 113	29
			Y	2 803 435	51	18
Merged	(No)	1	n	7 371 922	888 089	177 972
			Y	7 371 922	278 196	5939
		2	N	6 592 326	108 493	110
			Y	6 592 326	1688	42
	$E_{max} = 1$	1	N	6 728 314	410 712	75
			Y	6 728 314	22 443	39
		2	N	6 407 439	89 837	534
			Y	6 407 439	619	159

Reads that were predicted to be chimeric by UCHIME were discarded prior to this analysis (because chimera detection is a specialized task, and chimeras account for a large majority of the unique amplicon sequences, obscuring the underlying biological diversity). The reads contain 21 species, plus a few contaminants (Edgar, 2013). *Ab.* is minimum abundance (1 = all reads, 2 = singletons discarded), *Den.* is *n* (no denoising) or *Y* (UNOISE), *Reads* is the number of reads after processing, *Uniques* is number of unique read sequences after processing and OTUs is the number of clusters at 97% identity (made by running UCLUST on the unique sequences sorted by decreasing abundance).

PANDAseq merged 74% of these pairs with implied overlap lengths ranging from 2 to 26 bases. The other tested mergers generated no false positives on this dataset.

3.8 PANDAseq quality filtering

PANDAseq implements quality filtering by setting a minimum value for the geometric mean (t) of the posterior probabilities $p_i^{correct} = 1 - p_i$ that the base calls are correct. The default value of the threshold is $t = 0.6$, corresponding to $p = 0.4$ or $Q = 4$. The motivation for this measure of quality is not stated by the authors and was not clear to us, and we also noted that a threshold of $p = 0.4$ would allow many errors in low-quality reads. We therefore asked whether using a more stringent t threshold could achieve comparable filtering performance to $E_{max} = 1$. To answer this, we tuned t to retain as close as possible to the same number of reads as USEARCH with $E_{max} = 1$. On all three datasets, PANDAseq with the tuned threshold produced fewer reads with zero and one errors and more reads with two or more errors than USEARCH, showing that E_{max} is a superior measure of quality (Supplementary Fig. S3).

4 Discussion

We introduced the expected number of errors as measure of read quality. We showed that imposing a maximum value on expected errors is an effective quality filter. We showed that most current paired read mergers do not correctly calculate the posterior quality scores and that PANDAseq will align unrelated random pairs, potentially causing a high rate of false-positive merges in biological data.

We suggest that measurements of Q score accuracy and the correlation of E with the true error rate should be a standard step in any

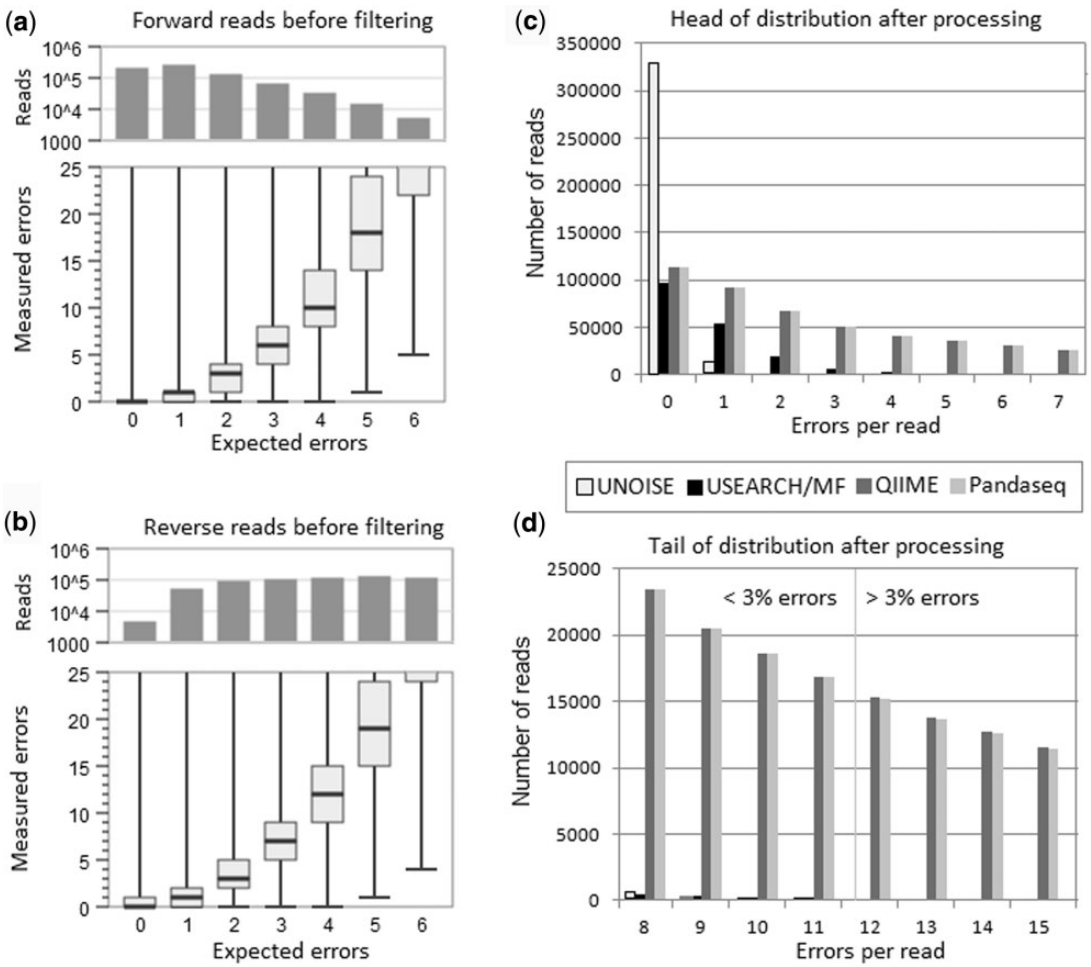

Fig. 3. Results on the MOCK1 dataset. For equivalent results for MOCK2 and PHIX, see Supplementary Figures S9 and S10. The box/whisker plots in panels (**a**) and (**b**) show the correlation between E and measured errors before filtering in the forward and reverse reads respectively. E is rounded to integers and binned so that, e.g. the bin for $E=2$ contains reads with $1.5 \leq E < 2.5$. For each bin, the top and bottom of the box indicates the upper and lower quartile, respectively, and the line inside the box indicates the median value. The upper and lower whiskers indicate the maximum and minimum measured errors, respectively. In all cases, the maximum value is >25 and is probably explained by a read that is a polymerase chain reaction (PCR) artifact, such as an unfiltered chimera, with true number of sequencing errors much less than 25. The upper histograms in panels (a) and (b) show the numbers of reads falling into each E bin. This shows that the reverse reads have more reads with lower quality, as is typically seen with Illumina sequencing. However, the correlation seen in the box/whisker plots appear similar between the forward and reverse reads, suggesting that the Q score accuracy is comparable. These results show that E tends to underestimate the number of errors for larger values of E. The histograms in panels (**c**) and (**d**) report the distribution after merging and filtering of the observed numbers of errors per read in the head (<3% errors) and tail (>3% errors), respectively, showing that $E_{max}=1$ allows most reads with no errors and a majority of reads with one error, and further dramatically reduces the frequency of reads in the tail compared with QIIME and PANDAseq

next-generation read analysis where Q scores are used, especially when coverage is low. This will enable informed setting of parameters such as the E_{max} threshold. With Illumina amplicon sequencing, a control sample such as a mock community would be ideal, but the additional cost and effort may be prohibitive. However, a PhiX spike-in is often added to amplicon libraries before Illumina sequencing (Kozich *et al.*, 2013), and our results here show that PhiX reads can function as an informative control sample for Q score analysis.

We suggest the default value $E_{max}=1$ as a filtering threshold as the most probable number of errors is zero for the filtered reads. More or less stringent thresholds may be appropriate to correct for biases in the Q scores or to make trade-offs between sensitivity and error rate in downstream analysis.

We believe that these methods will enable improved accuracy of biological inferences in a broad range of NGS experiments.

Conflict of Interest: none declared.

References

Bokulich,N.A. *et al.* (2013) Quality-filtering vastly improves diversity estimates from Illumina amplicon sequencing. *Nat. Methods*, **10**, 57–59.

Caporaso,J.G. *et al.* (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods*, **7**, 335–336.

Edgar,R.C. (2010) Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, **26**, 2460–2461.

Edgar,R.C. (2013) UPARSE: highly accurate OTU sequences from microbial amplicon reads. *Nat. Methods*, **10**, 996–998.

Edgar,R.C. *et al.* (2011) UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics*, **27**, 2194–2200.

Glenn,T.C. (2011) Field guide to next-generation DNA sequencers. *Mol. Ecol. Resour.*, **11**, 759–769.

Haas,B. *et al.* (2011) Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome Res.*, 494–504.

Huse,S.M. *et al.* (2010) Ironing out the wrinkles in the rare biosphere through improved OTU clustering. *Environ. Microbiol.*, **12**, 1889–1898.

Jiang,N. *et al.* (2013) Lineage structure of the human antibody repertoire in response to influenza vaccination. *Sci. Transl. Med.*, **5**, 171ra19.

Kozich,J.J. *et al*. (2013) Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq illumina sequencing platform. *Appl. Environ. Microbiol.*, **79**, 5112–5120.

Le Cam,L. (1960) An approximation theorem for the Poisson binomial distribution. *Pacific J. Math.*, **10**, 1181–1197.

Liu,B. *et al*. (2012) COPE: An accurate k-mer-based pair-end reads connection tool to facilitate genome assembly. *Bioinformatics*, **28**, 2870–2874.

Magoč,T. and Salzberg,S.L. (2011) FLASH: fast length adjustment of short reads to improve genome assemblies. *Bioinformatics*, **27**, 2957–2963.

Masella,A.P. *et al*. (2012) PANDAseq: paired-end assembler for illumina sequences. *BMC Bioinformatics*, **13**, 31.

Quince,C. *et al*. (2009) Accurate determination of microbial diversity from 454 pyrosequencing data. *Nat. Methods*, **6**, 639–641.

Quince,C. *et al*. (2011) Removing noise from pyrosequenced amplicons. *BMC Bioinformatics*, **12**, 38.

Reeder,J. and Knight,R. (2010) Rapidly denoising pyrosequencing amplicon reads by exploiting rank-abundance distributions. *Nat. Methods*, **7**, 668–669.

Reveillaud,J. *et al*. (2014) Host-specificity among abundant and rare taxa in the sponge microbiome. *ISME J.*, **8**, 1198–1209.

Rodrigue,S. *et al*. (2010) Unlocking short read sequencing for metagenomics. *PLoS One*, **5**.

Schloss,P.D. *et al*. (2011) Reducing the effects of PCR amplification and sequencing artifacts on 16S rRNA-based studies. *PLoS One*, **6**, e27310.

Wang,Y.H. (1993) On the number of successes in independent trials. *Stat. Sin.*, **3**, 295–312.

Zhang,J. *et al*. (2014) PEAR: a fast and accurate Illumina paired-end read merger. *Bioinformatics*, **30**, 614–620.