QUIKR: A METHOD FOR RAPID RECONSTRUCTION OF BACTERIAL COMMUNITIES VIA COMPRESSIVE SENSING

DAVID KOSLICKI, SIMON FOUCART, AND GAIL ROSEN

Abstract. Many metagenomic studies compare hundreds to thousands of environmental and health-related samples by extracting and sequencing their 16S rRNA amplicons and measuring their similarity using beta-diversity metrics. However, one of the first steps - to classify the operational taxonomic units withing the sample - can be a computationally time-consuming task since most methods rely on computing the taxonomic assignment of each individual read out of tens to hundreds of thousands of reads. We introduce Quikr: a QUadratic, K-mer based, Iterative, Reconstruction method which computes a vector of taxonomic assignments and their proportions in the sample using an optimization technique motivated from the mathematical theory of compressive sensing. On both simulated and actual biological data, we demonstrate that Quikr is typically more accurate as well as typically orders of magnitude faster than the most commonly utilized taxonomic assignment technique (the Ribosomal Database Project's Naïve Bayesian Classifier). Furthermore, the technique is shown to be unaffected by the presence of chimeras thereby allowing for the circumvention of the time-intensive step of chimera filtering. The Quikr computational package (using MATLAB or Octave) for the Linux and Mac platforms is available at http://sourceforge.net/projects/quikr/.

1. Introduction

Reconstructing the taxonomic composition of a bacterial community taken from an environmental sample (be it an ocean, soil, or human associated sample) is critical for understanding the role that such a community might play in affecting change in that environment. A popular reconstruction approach ([14], [27], [18], [10], [28]) is to utilize 16S rRNA amplicon sequencing (like Roche's 454 technology) to produce many ($\sim 400,000$ to $\sim 1,000,000$) moderate length (~ 400bp to ~ 700bp) reads of specific variable regions of the 16S rRNA gene and then individually classify these reads using a custom database with BLAST or in a Bayesian framework like the Ribosomal Database Project's (RDP) Naïve Bayesian Classifier (NBC) [28]. RDP's NBC is widely used due to its speed but it can still take several days to assign millions of reads on a desktop computer, thereby alienating users who do not have access to large computer clusters.

We introduce a method that enables desktop analysis: we take a novel approach by reconstructing all taxonomic concentrations of a bacterial community simultaneously (as opposed to read-by-read classification). This allows for orders of magnitude decrease in execution time while maintaining comparable (and often better) reconstruction fidelity. This method, based on ideas from compressive sensing, was inspired by and tangentially related to [2] wherein sparsity-promoting algorithms were utilized to analyze mixtures of dye-terminator reads resulting from Sanger sequencing. Here, however, we take a k-mer based approach that is designed for high-throughput sequencing technologies. Put briefly, our method measures the frequency of k-mers (for a fixed $k \sim 6$) in a database of 16S rRNA genes for known bacteria, calculates the frequency of k-mers in the given sample, and then reconstructs the concentrations of the bacteria in the sample by solving an underdetermined system of linear equations under a sparsity assumption. To solve this system, we employ

Date: February 27, 2013.

This work was initiated when D.K. was with Drexel University.

D.K. supported in part by the NSF under agreement No. 0931642.

Research supported by NSF grant DMS-1120622.

MATLAB's [1] iterative implementation of typical nonnegative least squares and hence we refer to this method as *Quikr*: QUadratic, Iterative, K-mer based Reconstruction. We point out that Quikr has not yet been optimized for performance but still demonstrates orders of magnitude speed improvement over RDP's NBC.

2. Methods

2.1. **k-mer Training Matrix.** The training step consists of converting an input database of 16S rRNA sequences into a *k-mer training matrix.* For a fixed k-mer size, we calculate the frequency of each k-mer in each database sequence. Hence, given a database of 16S rRNA sequences $D = \{d_1, \dots, d_M\}$, the $(i,j)^{\text{th}}$ entry of the k-mer training matrix $A^{(k)}$ is the frequency of the i^{th} k-mer (in lexicographic order) in the j^{th} sequence d_j.

Herein, we consider two different databases of 16S rRNA sequences. The first database, D_{small}, is the same as the training database for RDP's NBC version 7. This database consists of 10,046 sequences and will allow for direct comparison of Quikr to RDP's NBC.

The second database, D_{large}, consists of the 275,727 sequences that remained after applying TaxCollector [13] to the entire RDP 16S rRNA database 10.28. Applying TaxCollector had the net effect of labeling each sequence with taxonomic information obtained from NCBI ([24], [6]), discarding duplicate sequences, and discarding sequences that were missing genus labels. Training the RDP's NBC with this database would lead to prohibitively long classification times and so will demonstrate how Quikr can incorporate much more known information than RDP's NBC.

2.2. **Sample k-mer Frequencies.** Given a sample dataset of 16S rRNA reads, we calculate the frequency of all k-mers in the entire sample. We refer to this vector $s^{(k)}$ as the *sample k-mer frequency vector.* Note that the calculation of $s^{(k)}$ is an easily parallelizable problem that can be computed very efficiently in an online fashion.

2.3. **Sparsity Promoting Quadratic Optimization.** We assume that the given environmental sample only contains bacteria that exist in the database $D = \{d_1, \dots, d_M\}$ being utilized. Hence we can represent the composition of the sample as a vector x with nonnegative entries summing to one (i.e. a probability vector) where x_i is the concentration of the organism with 16S rRNA sequence d_i. However, as will be demonstrated in section 3.6, the Quikr method still performs well when the sample *does* contain novel bacteria not in the database being utilized.

We consider the idealized situation, in which sample noise and errors introduced by short reads are ignored. The problem at hand is then to reconstruct the bacterial concentrations x by solving the underdetermined linear system

$$A^{(k)}x = s^{(k)}. \tag{2.1}$$

Under the plausible assumption that relatively few bacteria from the database D are actually present in the given sample (that is, x is a sparse vector), we can solve equation (2.1) by modifying some techniques from compressive sensing. We use a variant of basis-pursuit denoising [8] which reduces to a nonnegative least squares problem. The details regarding this sparsity promoting, iterative, quadratic optimization procedure are contained in Appendix B.2.

Occasionally, Quikr experiences convergence issues. However, as detailed in Appendix B.2, filtering out the shortest sequences from a given sample solved this issue in every situation we encountered.

2.4. **Reconstruction Metrics.** There are a variety of metrics employed in the literature to asses bacterial community reconstruction fidelity (for example, see [25], [2], [9], [23], and [28]). We denote the *actual* and *predicted* concentrations of the bacteria as probability vectors x and x^* respectively. The reconstruction metric primarily employed herein is the ℓ_1 distance between x and x^*: $||x - x^*||_{\ell_1}$. This quantity takes values between 0 and 2 (with perfect reconstruction being

$||x-x^*||_{\ell_1}=0$) and is commonly referred to as "total error" (as it is the total of the absolute errors). We also use precision, sensitivity, specificity, and accuracy; these error metrics vary between 0 and 1 (with higher values reflecting better reconstruction fidelity) . The definitions of these quantities are contained in Appendix C.

Note that the correlation between x and x^* is not an effective reconstruction metric because the sparsity of x and x^* and the high number of true negatives typically make $\text{corr}(x, x^*) := x^\top x^*/(||x||_2||x^*||_2)$ too close to the optimal value 1.

The term *reconstruction fidelity* will be used to communicate generically how well x^* approximates x.

2.5. **Simulated Data.** To test the performance of the Quikr method, the shotgun/amplicon read simulator Grinder [3] was used to generate a large variety of simulated 454 pyrosequencing datasets. These simulated datasets have a wide range of differing characteristics designed to replicate Roche's FLX and FLX+ technologies in a variety of conditions (for example: differing species abundances, read coverages, read lengths, error models, abundance models, chimera percentages, etc.). The particular parameter values can be found in Appendix A. This resulted in 216 different simulated datasets with a total of over 172 million reads with average read lengths normally distributed around either 400bp or 700bp, resulting in over 78 billion bases.

2.6. **Mock Communities.** To benchmark the Quikr method on real biological data, we examined the mock microbial communities developed in [16]. These communities contain equivalent concentrations of 16S rRNA genes for each of 21 different organisms that span a diverse range of properties (GC content, genome size, etc.). This mock microbial community was then sequenced independently at four different institutions with primers designed to target the V1-V3, V3-V5, and V6-V9 variable regions for a total of 12 different 454 datasets with an average read length of 439bp and standard deviation 38bp. Details regarding the precise conditions under which this data was obtained appear in [16, pages 499-500].

2.7. **Human Microbiome Data.** To further benchmark the Quikr method on real biological data, we applied the Quikr method to the Human Microbiome Project's [26] trimmed sequences resulting from SRA study id SRP002395. This dataset consists of approximately 72 million reads over 5,034 samples targeting the V1-V2 and V6-V9 variables regions.

3. Results

3.1. **Speed Comparison.** We performed all benchmarks against RDP's NBC since this is considered to be the fastest 16S rRNA classifier to date [20]. Figure 1 shows a log-log plot of the number of reads analyzed versus time for RDP's NBC version 10.28 with training set 7 (this is the same as database D_{small}, see section 2.1) and Quikr with $k=6$ using the database D_{small}. Note the significant improvement in speed: it takes Quikr well less than 1 minute to analyze over 1 million reads. While RDP's NBC computational complexity in the number of reads N is approximately $\mathcal{O}(N)$, on this data Quikr is approximately $\mathcal{O}(N^{1/5})$.

3.2. **Simulated Data Results.** The Quikr method was applied to all 216 simulated datasets using k-mer sizes in the range $k=1,\dots,6$ for both databases D_{small} and D_{large}. We observed that at the genus level the mean ℓ_1 error decreased roughly linearly (linear regression $R^2=0.953$) as a function of k-mer size. However, the total algorithm time increased exponentially. This behavior is to be expected due to the exponential increase in number of k-mers as a function of k. These patterns were observed at all taxonomic ranks with both training databases. We recommend using the k-mer size $k=6$ as this provides a good trade-off between reconstruction fidelity and execution time. Parts (C) and (D) in figure 2 demonstrate that, as expected, the reconstruction error increases as

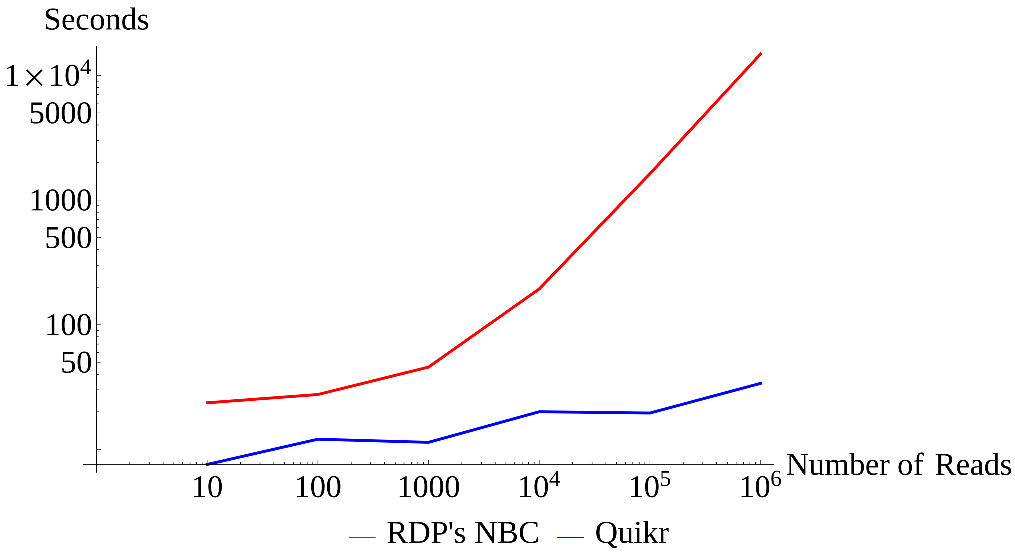


FIGURE 1. Log-Log plot of Number of Reads Versus Time (in seconds) for both RDP's NBC and Quikr.

one moves down the taxonomic ranks (e.g. reconstruction at the phylum level is better than at the genus level).

For comparison purposes, we also classified the simulated data using the popular Ribosomal Database Project's Naïve Bayesian classifier [28] version 10.28 with training set 7 (this is the same training data as the database D_{small}). Figure 2 compares the timing, mean ℓ_1 error at various taxonomic ranks, as well as precision, sensitivity, specificity, and accuracy at the genus level between Quikr (using k-mer size $k = 6$) and RDP's NBC.

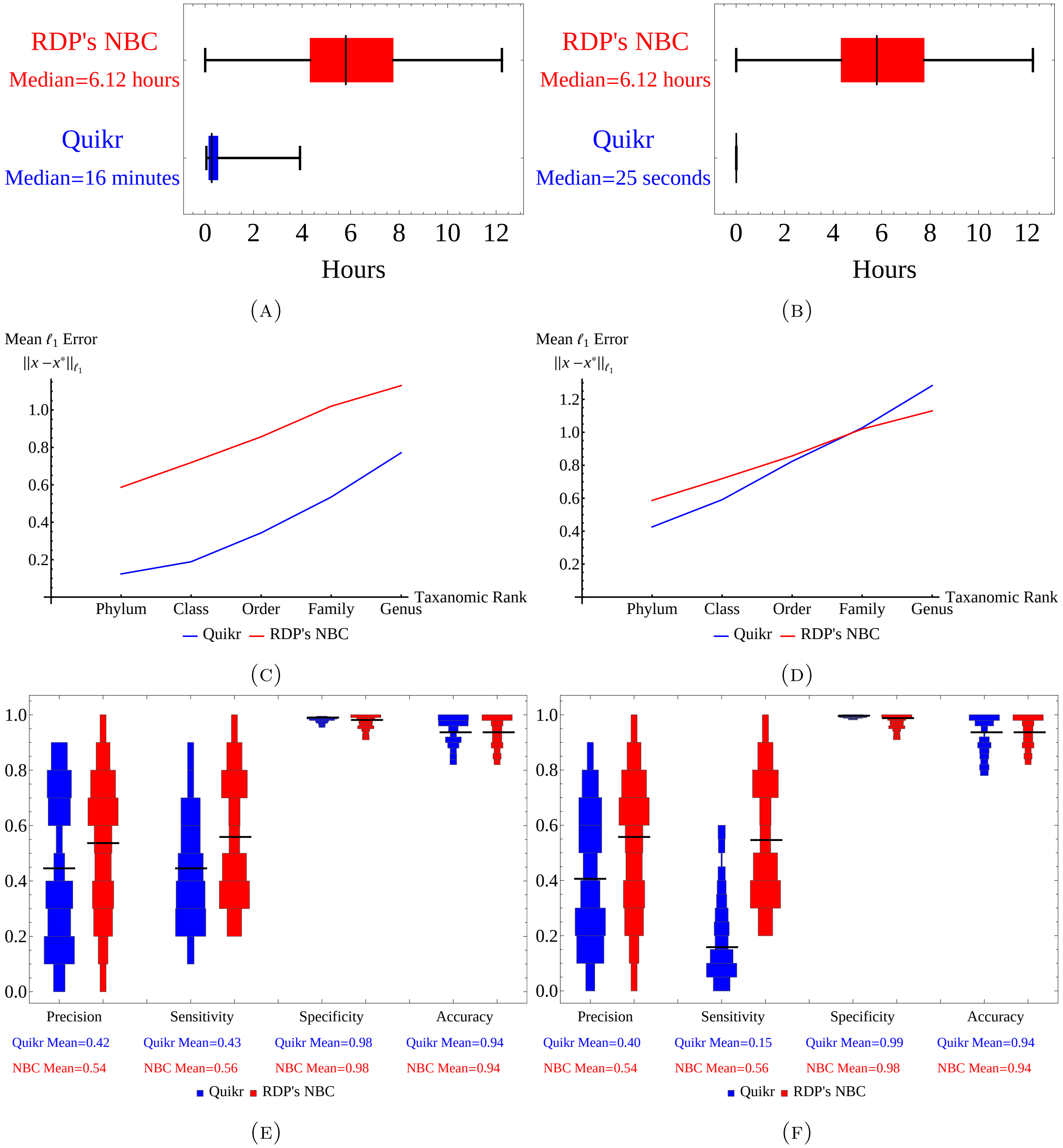
As parts (A) and (B) in figure 2 show, Quikr is orders of magnitude faster than RDP's NBC no matter which training database is used. Indeed, using D_{large}, Quikr took an average of 1730 seconds per dataset (or 520 reads per second). Using D_{small}, Quikr took an average of only 26.4 seconds per dataset (or 34,091 reads per second). Compare this to RDP's NBC taking an average of 23,978 seconds per dataset (or 38 reads per second).

Parts (C) and (D) in figure 2 demonstrate that both methods show an increase in mean ℓ_1 error as one moves to lower taxonomic ranks. At the genus level and using the training database D_{large}, Quikr shows a 46.5% improvement in ℓ_1 error over RDP's NBC. Using the training database D_{small}, Quikr has comparable error to RDP's NBC down to the family level. Using this smaller database, Quikr results in more error than RDP's NBC at the genus level.

Parts (E) and (F) in figure 2 show that when using D_{large}, Quikr has comparable specificity and accuracy, and only slightly lower averages for precision and sensitivity when compared to RDP's NBC at the genus level. This pattern continues when using the database D_{small} except here Quikr is much less sensitive than RDP's NBC but shows comparable precision, specificity, and accuracy.

These results demonstrate that when using the training database D_{small}, Quikr is an extremely fast method that gives a good high-level characterization of a given sample. When using the training database D_{large}, Quikr is a fast and very accurate classification method even down to the genus level.

3.3. **Mock Communities Results.** We analyzed these 12 mock communities with the Quikr method for k-mer size $k = 6$ with both training databases D_{large} and D_{small}, as well as the RDP's NBC version 10.28 with training set 7 (which is the same as database D_{small}). Figure 3 compares the timing, mean ℓ_1 error at various taxonomic ranks, as well as the remaining error metrics at the



(A)

(B)

(C)

(D)

(E)

(F)

FIGURE 2. Comparison of Quikr to RDP's NBC using the 216 simulated datasets. Throughout, RDP's NBC version 10.28 with training set 7 (i.e. D_{small}) was utilized. (A)-(B) Boxplot of algorithm execution time when Quikr was trained using D_{large} and D_{small} respectively. Whiskers denote range of the data, vertical black bars designate the median, and the boxes demarcate quantiles. (C)-(D) ℓ_1 error averaged over all 216 simulated datasets versus taxonomic rank when Quikr was trained using D_{large} and D_{small} respectively. (E)-(F) Histogram densities for other error metrics at the genus level when Quikr was trained using D_{large} and D_{small} respectively. The horizontal black bars represent the mean.

genus level between Quikr and RDP's NBC. Similarly to the simulated data in section 2.5, with training database D_{large}, Quikr is on average faster than RDP's NBC, and significantly faster when using the training database D_{small} (see parts (A) and (B) in figure 3).

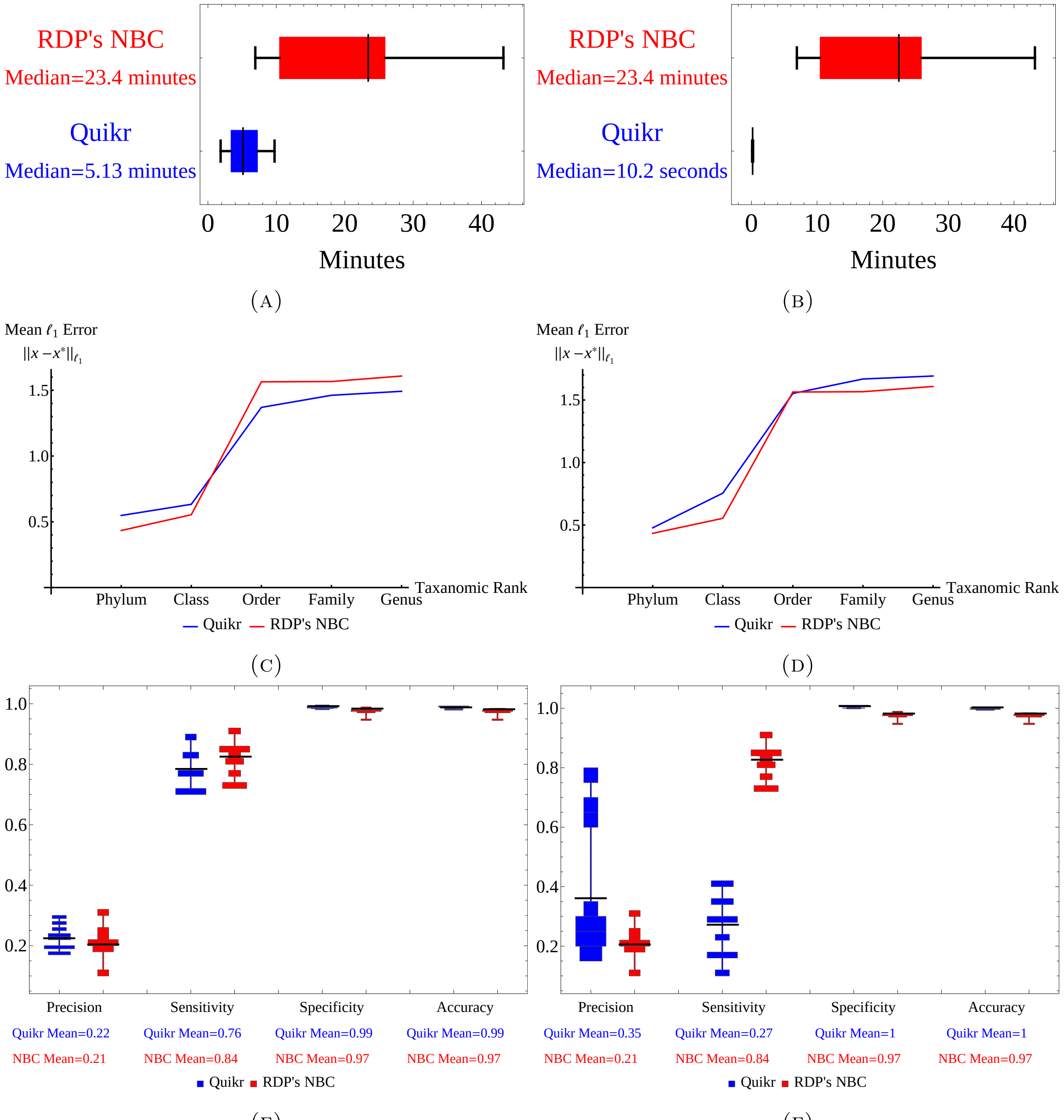
As parts (C) and (D) of figure 3 show, the ℓ_1 errors of both methods are comparable. Furthermore, when using the training database D_{large}, Quikr has less error than RDP's NBC at the genus level. Lastly, when Quikr uses the training database D_{large}, both methods have comparable precision, sensitivity, specificity, and accuracy (note Quikr is slightly more precise, specific, and accurate). When using D_{small}, Quikr is significantly less sensitive than RDP's NBC, but the other error metrics give similar values.

This demonstrates again that when using the training database D_{small}, Quikr is an extremely fast method that gives a good high-level characterization of a given sample. When using the training database D_{large}, Quikr is a fast and very accurate classification technique.

3.4. **Human Microbiome Project (HMP) Results.** To demonstrate that Quikr is fit for utilization on a desktop computer, we analyzed the 5,034 samples of HMP data on an iMac with a 3.4 GHz Intel i-7 processor. Utilizing the default training database D_{small} (which corresponds to RDP's training set 7), Quikr took 7.6 hours to analyze the entire HMP data set. Re-training with the Greengenes [11] 91%-OTU database of 5,878 sequences, Quikr took only 4.8 hours to analyze the entire HMP data set. The results of analyzing the HMP data with the Grenegenes database were then analyzed in QIIME [7] to produce a PCoA plot which is included in figure 4. This plot can be compared to figure 1a) in [15]. The results are similar enough in their clustering to conclude that Quikr is effective in facilitating the transformation of raw reads into an accurate PCoA plot in less than workday on a typical scientist's desktop computer.

3.5. **Chimeras.** The presence of chimeras in an amplicon sample can significantly affect downstream analysis when using classification algorithms such as Bayesian classifiers [4], and is possibly the culprit for over-estimates of the so-called "rare biosphere" [12]. Identifying and removing chimeras is a computationally intensive and only partially solved problem ([12], [16], [22], [17]). It is therefore a significant advantage of the Quikr method that it is completely unaffected by the presence of chimeras. Quikr's unaffectedness by chimeras is due to the k-mer frequency of a chimera being well-estimated by the weighted sum of the k-mer frequencies of the constituent sequences that generated the chimera. To present experimental evidence of this invariance, we selected Grinder [3] parameters to be the same as in section 3.6, but varied the percentage of chimeras from 0% to 100% in 10% increments, with 10 simulations being performed at each increment. An ANOVA analysis resulted in a p-value of $p = .927$, hence there is no statistically significant evidence that the slope of a linear regression deviates from zero. Figure 5 illustrates this fact by plotting the mean ℓ_1 error and standard deviation over the 10 simulations versus percent chimeras. Hence, it can be concluded that it is unnecessary to filter for chimeras before using the Quikr method.

3.6. **Cross-Validation.** To gauge how well the Quikr method will perform when the given sample contains 16S rRNA not in the database (simulating novelty), we performed a 5-fold cross-validation. Throughout the cross-validation, the k-mer size was fixed at $k = 6$. The database D_{large} described in section 2.1 was partitioned into 5 disjoint sets and $1/5^{\text{th}}$ was set aside as testing data with the remaining $4/5^{\text{ths}}$ used to form a new k-mer matrix as in section 2.1. Grinder [3] parameters were then chosen to generate a test sample from the testing data. In particular, these parameters were chosen as follows: primers targeting the V1-V3 variable regions, read lengths normally distributed with mean 400bp and standard deviation 50bp, 800,000 total reads, exponential abundance model, diversity of 100 species, homopolymer error model as in Balzer [5], and 10% chimera percentage. The mean of each reconstruction metrics was then taken over the choice of which $1/5^{\text{th}}$ was the testing data. Lastly, an average was taken over 10 iterates of this procedure. RDP's NBC was also utilized to classify the Grinder test samples.



(A)

(B)

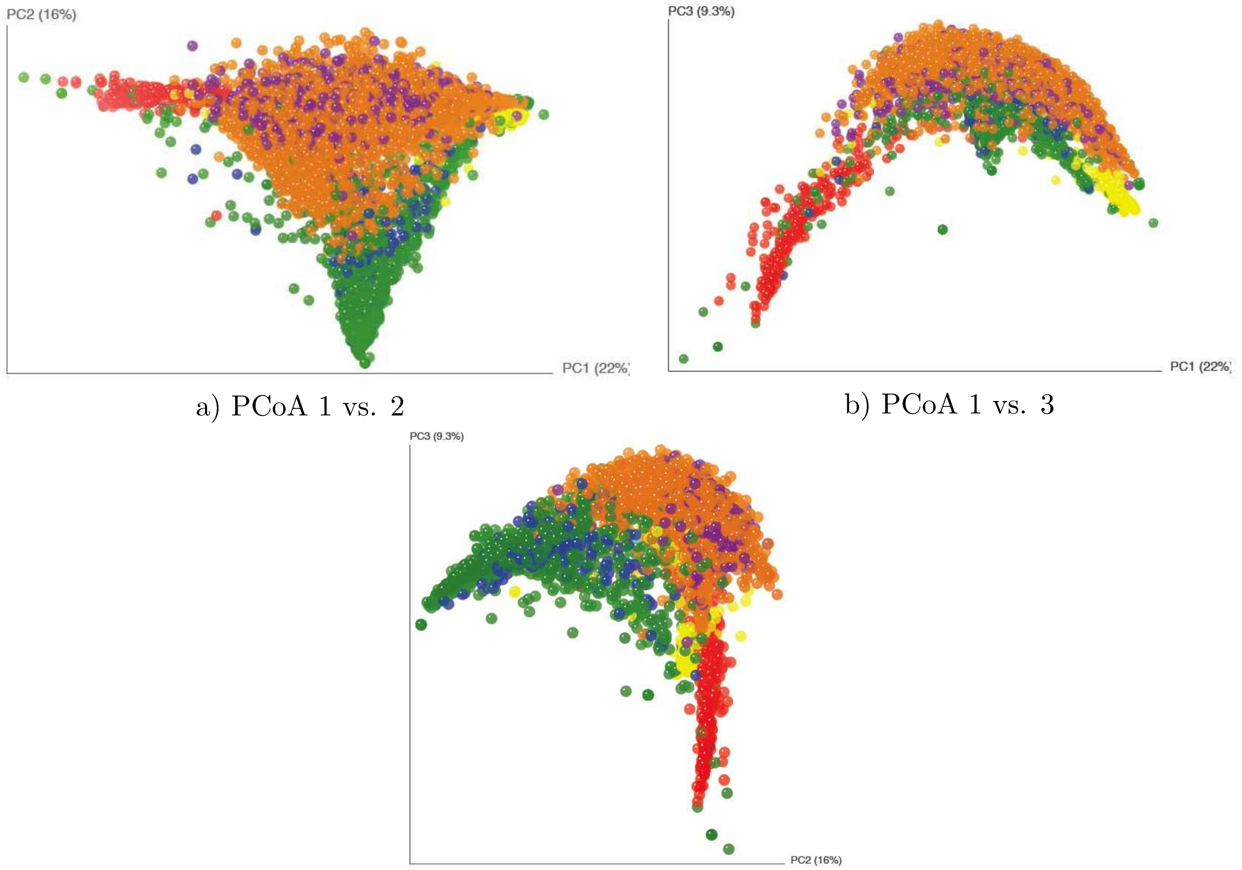
(C)

(D)

(E)

(F)

FIGURE 3. Comparison of Quikr to RDP's NBC using the mock communities. Throughout, RDP's NBC version 10.28 with training set 7 (i.e. D_{small}) was utilized. (A)-(B) Boxplot of algorithm execution time when Quikr was trained using D_{large} and D_{small} respectively. Whiskers denote range of the data, vertical black bars designate the median, the boxes demarcate quantiles. (C)-(D) ℓ_1 error averaged over all the mock communities versus taxonomic rank when Quikr was trained using D_{large} and D_{small} respectively. (E)-(F) Histogram densities for other error metrics at the genus level when Quikr was trained using D_{large} and D_{small} respectively. The horizontal black bars represent the mean.



a) PCoA 1 vs. 2

b) PCoA 1 vs. 3

c) PCoA 2 vs. 3

FIGURE 4. QIIME (weighted Unifrac) analysis performed using the Greengenes 91% OTU database, which took around 6 hours for Quikr+QIIME complete analysis. Color legend: Gut (Red), Oral (Orange), Throat (Purple), Skin (Green), Nasal (Blue), and Urogenital (Yellow).

TABLE 1. Results of 10 Iterates of the 5-fold Cross-Validation Procedure at the Genus Level (smaller values are better)

	Quikr	RDP's NBC
Mean ℓ_1 error ± variance	0.835 ± 0.00354	1.209 ± 0.0792

Table 1 summarizes the results of this procedure for the ℓ_1 error metric. Since Quikr has a smaller mean ℓ_1 error and tighter variance, this demonstrates that even if the given sample contains novel sequences not present in the database, the Quikr method will still give high reconstruction fidelity down to the genus level. Similar results were observed for the remaining error metrics.

4. DISCUSSION

Quikr represents a new paradigm in algorithms for bacterial community reconstruction. By leveraging ideas from compressive sensing, an entire sample can be analyzed quickly and accurately. Depending on how it is trained, Quikr can be used as either an extremely rapid, almost constant time, high-level community profiling tool or else (using a larger training database) a fast, extremely accurate technique. Besides improvements in speed, other advantages include the ability to utilize

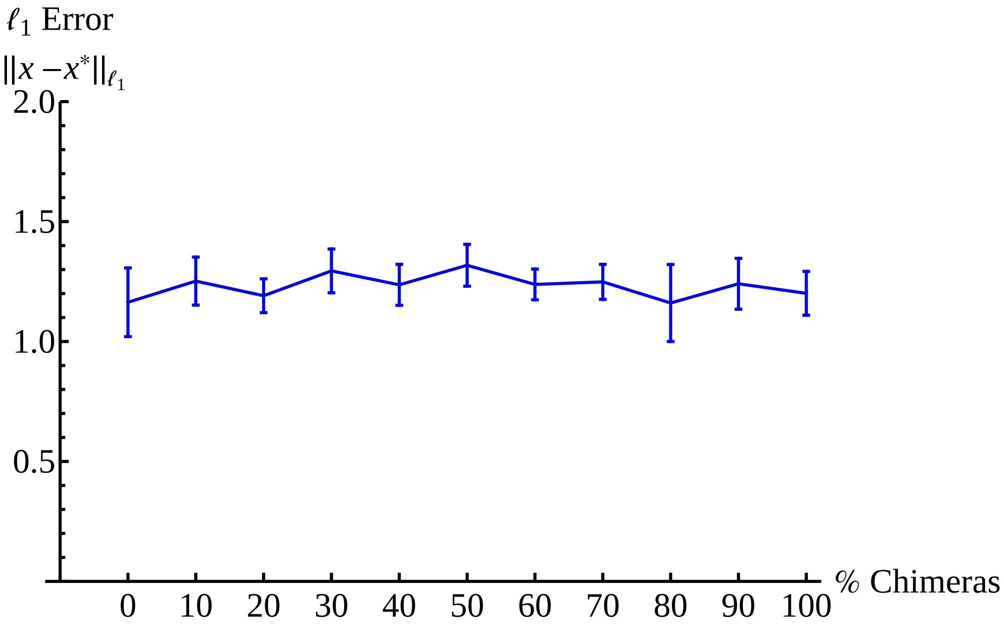


Figure 5. Mean ℓ_1 error at the genus level for the Quikr method versus percentage of chimeras. Error bars depict standard deviation over 10 simulations.

massive training databases (like D_{large}) that would be much too large for standard techniques (like RDP's NBC). Furthermore, Quikr is unaffected by the presence of chimeras, so the time-consuming chimera-removal step in standard analytic pipelines can be completely circumvented.

The Quikr computational package (using MATLAB or Octave) for the Linux and Mac platforms is available at http://sourceforge.net/projects/quikr/.

Appendix A. Design of Grinder Experiment

We detail here the production of the simulated data produced by the shotgun/amplicon read simulator Grinder [3]. These datasets were designed to mimic reads generated by Roche's GS FLX and FLX+ amplicon systems, and hence we set the read-length distributions to be normally distributed with a mean of 400bp or 700bp and a standard deviation of 50bp or 100bp. We chose the primers B27F, B357F, and BU968F to target the V1-V3, V3-V5, and V6-V9 variable regions respectively. Only forward primers were utilized since amplicon sequencing allows for filtering on sequencing direction. Three different diversity models were chosen to be $10^2, 10^3$, and 10^4 and abundance was modeled by a power-law or exponential distribution with parameters 0.705 and 1 respectively. Since most sequencing errors in these systems are due to homopolymer errors, we modeled such errors by using Balzer's [5] model of homopolymer error distribution. Chimera percentages were set at $0\%, 10\%$, and 30%. Since we consider only amplicon sequencing, no copy or length bias was employed.

Appendix B. Quikr Method Technical Details

B.1. **Mathematical Formulation.** Given the alphabet $\mathcal{A} = \{A, C, T, G\}$, let $\mathcal{A}^n$ denote the set of all words v of length $|v| = n$ on $\mathcal{A}$, and let $\mathcal{A}^* = \bigcup_{n \geq 0} \mathcal{A}^n$ be the set of all finite words on $\mathcal{A}$. Given a database $D = \{d_1, \dots, d_M\}$ of sequences $d_i \in \mathcal{A}^*$ and a set $S = \{s_1, \dots, s_t\}$ of sample sequences (the reads to be classified), we assume that for each s_l there is a unique j with $s_l = d_j$. This uniqueness is justified by the use of sequencing the highly variable 16S rRNA. We also make the assumption that the composition of the bacterial community is represented by the probability vector $x \in \mathbb{R}^M$ satisfying

$$x_i = \frac{1}{t} \sum_{l=1}^{t} \quad {}_{d_i}(s_l), \quad i = 1, \dots, M, \tag{B.1}$$

where $\quad {}_{d_i}(s_l)$ equals 1 if $s_l = d_i$ and 0 otherwise. This approximation is reasonable when the sample sequences are numerous and well distributed (i.e. for samples with high enough coverage, in the biological sense).

Fix a k-mer size and endow $\mathcal{A}^k = \{v_1, \dots, v_{4^k}\}$ with the lexicographic order. Let $\mathrm{occ}_v(w)$ represent the number of occurrences (with overlap) of the subword v in the word w. That is, for $w, v \in \mathcal{A}^n$, let

$$\mathrm{occ}_v(w) = |\{j : w_j w_{j+1} \cdots w_{j+|v|-1} = v\}|. \tag{B.2}$$

For $j = 1, \dots, M$ and $i = 1, \dots, 4^k$, define the *k-mer training matrix* entrywise as

$$A^{(k)}_{i,j} = \frac{occ_{v_i}(d_j)}{|d_j| - k + 1}. \tag{B.3}$$

The matrix $A^{(k)}$ satisfies $A^{(k)}_{i,j} \geq 0$ and is column-normalized, i.e.

$$\sum_{i=1}^{4^k} A^{(k)}_{i,j} = 1 \quad \text{for all } j = 1, \dots, M. \tag{B.4}$$

Define the *sample k-mer frequency vector* entrywise for $i = 1, \dots, 4^k$ as

$$s^{(k)}_i = \frac{1}{t} \sum_{l=1}^{t} \frac{\mathrm{occ}_{v_i}(s_l)}{|s_l| - k + 1}. \tag{B.5}$$

Our two assumptions imply that

$$A^{(k)} x = s^{(k)}. \tag{B.6}$$

We will try to recover the probability vector x satisfying $x_j \geq 0$ for all $j = 1, \dots, M$ and $\sum_{j=1}^{M} x_j = 1$ from information in the form of equation (B.6). Given that a bacterial community is typically distributed as a sparse vector x (a small percentage of all extant bacteria are actually present in a given sample), we pursue sparsity-promoting minimizations involving the ℓ_1-norm. In particular, we consider the following optimization problems.

$$\text{(BP)} \qquad \underset{z \in \mathbb{R}^M}{\text{minimize}} \; ||z||_1 \qquad \text{subject to } A^{(k)} z = s^{(k)},$$

$$(\text{BP}_{\geq 0}) \qquad \underset{z \in \mathbb{R}^M}{\text{minimize}} \; ||z||_1 \qquad \text{subject to } A^{(k)} z = s^{(k)} \text{ and } z \geq 0,$$

$$(\text{REG}_1^2) \qquad \underset{z \in \mathbb{R}^M}{\text{minimize}} \; ||z||_1^2 + \lambda^2 ||A^{(k)} z - s^{(k)}||_2^2 \qquad \text{subject to } z \geq 0,$$

It can be demonstrated, thanks to (B.4), that (BP) and ($\text{BP}_{\geq 0}$) are equivalent in the sense that x is a solution of (BP) if and only if it is a solution of ($\text{BP}_{\geq 0}$), and that the latter is approached by solutions of (REG_1^2) when $\lambda \to \infty$.

We shall solve (REG_1^2) since it has the notable advantage of being transformed into a nonnegative least squares problem. Indeed, with

$$\tilde{A}^{(k)} := \left[\frac{1 \cdots 1}{\lambda A^{(k)}} \right], \qquad \tilde{s}^{(k)} := \left[\frac{0}{\lambda s^{(k)}} \right],$$

the minimization (REG_1^2) is equivalent to

$$\text{(NNLSQ)} \qquad \underset{z\in\mathbb{R}^M}{\text{minimize}} \; ||\tilde{A}^{(k)}z - \tilde{s}^{(k)}||_2^2 \qquad \text{subject to } z \geq 0.$$

B.2. **Algorithmic Implementation.** To solve (NNLSQ) we utilized MATLAB's [1] implementation of `lsqnonneg()` which in turn is an implementation of the iterative algorithm described in [19]. Throughout, we used $\lambda = 10,000$. We did observe that when using D_{large} to form $A^{(k)}$, for some inputs $s^{(k)}$ the algorithm took much longer to converge as it iterated through the inner loop (see [19, page 161, (23.10)]). This is most likely due to many of the columns of $A^{(k)}$ being highly correlated (as our database was unfiltered, there remained many highly similar strains of the same species). However, since the algorithm execution time was still quicker or comparable to RDP's NBC, we do not consider this a disadvantage. Furthermore, one can simply increase the tolerance in `lsqnonneg()` or limit the number of iterations to speed convergence. We set the number of iterations of the inner loop to be at most 10,000. Furthermore, the inclusion of short sequences in a given sample can cause convergence issues. Removing these sequences (say, every sequence shorter than two standard deviations away from the mean sequence length in a sample) solved this convergence issue in every case we encountered.

To calculate the matrices $A^{(k)}$ and the vector $s^{(k)}$ we used a custom SML [21] subword counting program written by Christopher Cramer and compiled for Linux using MLton [29]. To further speed the calculation of the sample k-mer frequency vector $s^{(k)}$, we took advantage of the following approximation:

$$\hat{s}_i^{(k)} := \frac{\sum_{j=1}^{t} occ_{v_i}(s_j)}{\sum_{l=1}^{4^k}\sum_{j=1}^{t} occ_{v_l}(s_j)} = \frac{\sum_{j=1}^{t} occ_{v_i}(s_j)}{\sum_{l=1}^{t} |s_l| - k + 1} \approx \frac{1}{t}\sum_{j=1}^{t} \frac{occ_{v_i}(s_j)}{|s_j| - k + 1} = s_i^{(k)}$$

where the last approximation is true provided the lengths $|s_j|$ are reasonably similar (which they are for Roche's 454 technology).

All computations were performed on a single cluster of 32 Intel Xeon E7-4820 CPU's at 2.00GHz.

Appendix C. Assessment of Reconstruction

There are a variety of metrics employed in the literature to asses bacterial community reconstruction fidelity (for example, see [25], [2], [9], [23], [28]). We detail some of them here. Throughout the following, $x \in \mathbb{R}^M$ represents the probability vector (i.e. $x_i \geq 0$ and $\sum_{i=1}^{M} x_i = 1$) of *true* bacterial concentrations and $x^* \in \mathbb{R}^M$ represents the probability vector of *estimated* bacterial concentrations. The ℓ_1 error (or total error) is defined by

$$||x - x^*||_{\ell_1} = \sum_{i=1}^{M} |x_i - x_i^*|.$$

The ℓ_1 error ranges between 0 and 2, with perfect reconstruction for $||x - x^*||_{\ell_1} = 0$.

The following reconstruction metrics all depend on a notion of true/false positive and true/false negative. Thus a threshold must be defined as to what is considered to be a "true zero". A threshold was set at 10^{-4}; hence if an entry x_i or x_i^* is less than 10^{-4}, it is considered to be a zero. We introduce the approximate support of a probability vector $x \in \mathbb{R}^M$ as $\text{supp}(x) = \{i \text{ s.t. } x_i > 10^{-4}\}$.

For the following, we suppress the dependence on x and x^* for notational simplicity and use a superscript c to denote set complement. True/false positives and true/false negatives are defined as

$$\begin{aligned}
TP &= |supp(x) \ \cap supp(x^*) \ |, \\
FP &= |supp(x)^c \cap supp(x^*) \ |, \\
TN &= |supp(x)^c \cap supp(x^*)^c|, \\
FN &= |supp(x) \ \cap supp(x^*)^c|,
\end{aligned}$$

respectively. Accuracy, precision, sensitivity, and specificity are now defined as:

$$\begin{aligned}
\text{Precision} &= \frac{TP}{TP+FP}, \\
\text{Sensitivity} &= \frac{TP}{TP+FN}, \\
\text{Specificity} &= \frac{TN}{TN+FP}, \\
\text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN}.
\end{aligned}$$

Appendix D. Read-by-read Analysis

It is possible to adapt the Quikr method to a read-by-read analysis. However, the execution time for the Quikr method depends predominantly on the k-mer size and not on the number of reads in the given sample. Implemented in a read-by-read fashion (using each read to generate a sample k-mer frequency vector $s^{(k)}$), Quikr analyzes approximately 2 reads per second for k-mer size $k = 6$ when using the training database D_{small}. This currently unoptimized approach was used to re-analyze the mock communities described in the main text. Compared to the whole-sample approach detailed in the main text's Methods section, the read-by-read implementation demonstrated almost identical error profiles: averaging over all error metrics at the genus level, there was only a 5.72% improvement with the read-by-read versus the whole-sample method. However, the read-by-read analysis was on average 17,046 times slower and so on par with RDP's NBC. It is expected that future optimization of the read-by-read implementation will allow this approach to become viable.

References

[1] MATLAB 2012b, The MathWorks, Inc., Natick, MA, USA.

[2] Amir, A. and Zuk, O. (2011). Bacterial community reconstruction using compressed sensing. *Journal of computational biology*, **18**(11), 1723–41.

[3] Angly, F. E., Willner, D., Rohwer, F., Hugenholtz, P., and Tyson, G. W. (2012). Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic acids research*, **61**(0), 1–8.

[4] Ashelford, K. E., Chuzhanova, N. A., Fry, J. C., Jones, A. J., and Weightman, A. J. (2005). At Least 1 in 20 16S rRNA Sequence Records Currently Held in Public Repositories Is Estimated To Contain Substantial Anomalies. *Applied Environmental Biology* **71**(12), 7724–7736.

[5] Balzer, S., Malde, K., Lanzén, A., Sharma, A., and Jonassen, I. (2010). Characteristics of 454 pyrosequencing data–enabling realistic simulation with flowsim. *Bioinformatics (Oxford, England)*, **26**(18), i420–5.

[6] Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Sayers, E. W. (2009). GenBank. *Nucleic acids research*, **37**(Database issue), D26–31.

[7] Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., Fierer, N., Pena, A. G., Goodrich, J. k., Gordon, J. I., Huttley, G. A., Kelley, S. T., Knights, D., Koenig, J. E., Ley, R. E., Lozupone, C. A., McDonald, D., Muegge, B. D. Pirrung, M., Reeder, J., Sevinsky, J. R., Turnbaugh, P. J., Walters, W. A., Widmann, J., Yatsunenko, T., Zaneveld, J., Knight, R. (2010). QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*, **7**, 335-336.

[8] Chen, S. S., Donoho, D. L., and Saunders, M. A. (1998). Atomic Decomposition by Basis Pursuit. *SIAM Journal on Scientific Computing*, **20**(1), 33–61.

[9] Clemente, J. C., Jansson, J., and Valiente, G. (2011). Flexible taxonomic assignment of ambiguous sequencing reads. *BMC bioinformatics*, **12**(1), 8.

[10] Cole, J. R., Wang, Q., Cardenas, E., Fish, J., Chai, B., Farris, R. J., Kulam-Syed-Mohideen, A. S., McGarrell, D. M., Marsh, T., Garrity, G. M., and Tiedje, J. M. (2009). The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic acids research*, **37**(Database issue), D141–5.

[11] DeSantis, T. Z., Hugenholtz, P., Larsen, N., Rojas, M., Brodie, E. L., Keller, K., Huber, T., Dalevi, D., Hu, P., and Andersen, G. L. (2006). Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Applied Environmental Microbiology* **75**, 5069–72.

[12] Edgar, R. C., Haas, B. J., Clemente, J. C., Quince, C., and Knight, R. (2011). UCHIME improves sensitivity and speed of chimera detection. *Bioinformatics (Oxford, England)*, **27**(16), 2194–200.

[13] Giongo, A., Davis-Richardson, A. G., Crabb, D. B., and Triplett, E. W. (2010). TaxCollector: Modifying Current 16S rRNA Databases for the Rapid Classification at Six Taxonomic Levels. *Diversity*, **2**(7), 1015–1025.

[14] Jumpstart Consortium Human Microbiome Project Data Generation Working Group (2012). Evaluation of 16S rRNA-Based Community Profiling for Human Microbiome Research. *PLoS ONE*, **7**(6), e39315.

[15] Koren, O., Knights, D., Gonzales, A., Waldron, L., Segat, N., Kight, R., Huttenhower, C., and Ley, R. E. A guide to enterotypes across the human body: Meta-analysis of microbial community structures in human microbiome datasets. *PLoS Computational Biology* **9**(1), e1002863

[16] Haas, B. J., Gevers, D., Earl, A. M., Feldgarden, M., Ward, D. V., Giannoukos, G., Ciulla, D., Tabbaa, D., Highlander, S. K., Sodergren, E., Methé, B., DeSantis, T. Z., Petrosino, J. F., Knight, R., and Birren, B. W. (2011). Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons. *Genome research*, **21**(3), 494–504.

[17] Huber, T., Faulkner, G., and Hugenholtz, P. (2004). Bellerophon: a program to detect chimeric sequences in multiple sequence alignments. *Bioinformatics (Oxford, England)*, **20**(14), 2317–9.

[18] Lan, Y., Wang, Q., Cole, J. R., and Rosen, G. L. (2012). Using the RDP classifier to predict taxonomic novelty and reduce the search space for finding novel organisms. *PLoS one*, **7**(3), e32491.

[19] Lawson, C. and Hanson, R. (1987). *Solving Least Squares Problems*. Prentice-Hall.

[20] Liu, Z., DeSantis, T.Z., Andersen, G.L., and Knight, R. Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers. *Nucleic Acids Res.* **38**(18), e120.

[21] Milner, R., Tofte, M., and Harper, R. (1997). *The Definition of Standard ML (Revised)*. MIT press, Cambridge, MA.

[22] Quince, C., Lanzen, A., Davenport, R. J., and Turnbaugh, P. J. (2011). Removing noise from pyrosequenced amplicons. *BMC bioinformatics*, **12**(1), 38.

[23] Rosen, G., Garbarine, E., Caseiro, D., Polikar, R., and Sokhansanj, B. (2008). Metagenome fragment classification using N-mer frequency profiles. *Advances in bioinformatics*, **2008**, 205969.

[24] Sayers, E. W., Barrett, T., Benson, D. A., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., DiCuccio, M., Edgar, R., Federhen, S., Feolo, M., Geer, L. Y., Helmberg, W., Kapustin, Y., Landsman, D., Lipman, D. J., Madden, T. L., Maglott, D. R., Miller, V., Mizrachi, I., Ostell, J., Pruitt, K. D., Schuler, G. D., Sequeira, E., Sherry, S. T., Shumway, M., Sirotkin, K., Souvorov, A., Starchenko, G., Tatusova, T. A., Wagner, L., Yaschenko, E., and Ye, J. (2009). Database resources of the National Center for Biotechnology Information. *Nucleic acids research*, **37**(Database issue), D5–15.

[25] Segata, N., Waldron, L., Ballarini, A., Narasimhan, V., Jousson, O., and Huttenhower, C. (2012). Metagenomic microbial community profiling using unique clade-specific marker genes. *Nature methods*, (9), 811–8147.

[26] The Human Microbiome Consortium. A framework for human microbiome research. *Nature* **486**, 215–221.

[27] Wang, C. and Zhang, D. (2011). A novel compression tool for efficient storage of genome resequencing data. *Nucleic acids research*, **39**(7), 5–10.

[28] Wang, Q., Garrity, G. M., Tiedje, J. M., and Cole, J. R. (2007). Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy. *Applied and environmental microbiology*, **73**(16), 5261–7.

[29] Weeks, S. (2006). Whole-program compilation in MLton. In *Proceedings of the 2006 workshop on ML*, page 1, New York, NY. ACM.

Mathematical Biosciences Institute, The Ohio State University, Columbus, OH 43201
E-mail address: `koslicki.1@mbi.osu.edu`

Department of Mathematics, Drexel University, Philadelphia, PA 19104

Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104