BIOINFORMATICS
Doc-Start
Vol. 00 no. 00 2015
Pages 1–7

ARK: Aggregation of Reads by K-means for Estimation of Bacterial Community Composition

Saikat Chatterjee [1]*, Damon Shahrivar [1], David Koslicki [2], Alan W. Walker [3], Suzanna C. Francis[4], Louise J. Fraser[5], Mikko Vehkaperä [6], Yueheng Lan [7], Jukka Corander [8]

[1]Dept of Communication Theory, KTH Royal Institute of Technology, Sweden
[2]Dept of Mathematics, Oregon State University, Corvallis, USA
[3]Microbiology Group, Rowett Institute of Nutrition and Health, University of Aberdeen, UK
[4]MRC Tropical Epidemiology Group, London School of Hygiene and Tropical Medicine, London, UK
[5]Illumina Cambridge Ltd., Chesterford Research Park, Essex, UK
[6]Dept of Signal Processing, Aalto University, Finland
[7]Dept of Physics, Tsinghua University, Beijing, China
[8]Dept of Mathematics and Statistics, University of Helsinki, Finland

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Estimation of bacterial community composition from high-throughput sequenced 16S rRNA gene amplicons is a key task in microbial ecology. Since the sequence data from each sample typically consist of a large number of reads and are adversely impacted by different levels of biological and technical noise, accurate analysis of such large datasets is challenging. Existing machine learning-based high fidelity estimation methods are typically incapable of handling a large dataset of sequence reads.

Results: There has been a recent surge of interest in using compressed sensing inspired methods to solve the estimation problem for bacterial community composition. These methods typically rely on summarizing the sequence data by frequencies of low-order k-mers and matching this information statistically with a taxonomically structured database. Here we show that the accuracy of the resulting community composition estimates can be substantially improved by aggregating the reads from a sample with an unsupervised machine learning approach prior to the estimation phase. In our approach we use a standard K-means clustering algorithm that partitions a large set of reads into subsets with reasonable computational cost to provide several vectors of first order statistics instead of only single summarization in terms of k-mer frequencies. The output of the clustering is then processed further using sparse signal processing to obtain the final estimate for each sample. The resulting method is called Aggregation of Reads by K-means (ARK), and it is based on a statistical argument via mixture density formulation. ARK is found to have higher fidelity and robustness than several recently introduced methods, with only a modest increase in computational complexity.

Availability: An open source, platform-independent implementation of the method in the Julia programming language is freely available at https://github.com/dkoslicki/ARK. A Matlab implementation is available at http://www.ee.kth.se/ctsoftware.

1 INTRODUCTION

The advent of high-throughput sequencing technologies has enabled detection of bacterial community composition at an unprecedented level of detail. A technological approach is to produce for each sample a large number of reads from amplicons of the 16S rRNA gene, which enables an identification and comparison of the relative frequencies of different taxonomic units present across samples. The rapidly increasing number of reads produced per sample results in the need for fast taxonomic classification of samples. This problem has attracted considerable recent attention [21, 14, 11, 16, 7].

Many existing approaches to the bacterial community composition estimation problem use 16S rRNA gene amplicon sequencing where a large amount of moderate length reads (around 250-500 bp) are produced from each sample and then generally either clustered or classified to obtain a composition estimate of taxonomic units. In the clustering approach, reads are grouped into taxonomic units by either distance-based or probabilistic methods [3, 9, 6], such that the actual taxonomic labels are assigned to the clusters afterward by matching their consensus sequences to a reference database. In contrast to the clustering methods, the classification approach is based on using a reference database directly to assign reads to meaningful biological units. Methods for the classification of reads have been based either on homology using sequence similarity, or on genomic signatures in terms of k-mer composition. Examples of homology-based methods include MEGAN [10, 15] and phylogenetic analysis [20]. Another popular approach is to use a Bayesian classifier [21, 18, 19]. One such method, the Ribosomal Database Project's (RDP) naïve Bayesian classifier (NBC) [21], assigns a label explicitly to each read produced for a particular sample. Despite the computational simplicity of NBC, the RDP classifier may still require several days to process a data set in a desktop environment. Given this challenge, considerably faster methods based on reconstructing mixtures of k-mer counts have been developed, for example, Taxy [14], Quikr [11] and recently

*To whom correspondence should be addressed. Email: sach@kth.se our proposed SEK [4]. SEK and Quikr are sparse signal processing based methods (based on compressed sensing), and SEK was shown to perform better than Quikr and Taxy in [4].

Taxy, Quikr and SEK all use as their main input a (statistical) mean vector of sample k-mer counts computed from the reads obtained for a sample. The necessary modeling assumption is that the sample mean vector of k-mer counts is sufficiently informative about the sample composition. These three methods do not use the reads in any additional way once the mean vector of k-mers is computed. We propose here an alternative basis of information aggregation that remains computationally tractable to allow processing of large sets of reads. Borrowing ideas from source coding in signal processing [13**?** , 5], clustering in machine learning, and divide-and-conquer based shotgun sequence assembly [17] our novel approach first segregates the k-mers into subsets, computes the mean vector for each subset, employs a standard method (such as Taxy, Quikr or SEK) to estimate composition for each subset, and finally fuses these estimates into a composition estimate jointly for all the reads. To segregate the k-mers into subsets, we employ the K-means clustering algorithm [8]. Since the K-means clustering algorithm is simple and computationally inexpensive for a reasonable number Q of clusters (subsets), it can be used to partition even fairly large sets of reads into more (intra)homogeneous subsets. By its very algorithmic definition, K-means clustering partitions the feature space into Q non-overlapping regions and provides a set of corresponding mean vectors. This is called *codebook generation* in vector quantization [13], originally from signal processing, coding and clustering. Our new method is termed as Aggregation of Reads by K-means (ARK). From the statistical perspective, theoretical justification of ARK stems from a modeling framework with a mixture of densities.

2 METHODS

2.1 Summarizing read sequence data by single mean k-mer counts

In the method description, we denote the non-negative real line by $\mathbb{R}_+$ and statistical expectation operator by $\mathbb{E}[.]$. First, we describe the previously published approach of using single k-mer summaries for each sample. Let $\mathbf{x} \in \mathbb{R}_+^{4^k}$ and $\mathcal{C}_m$ denote random k-mer feature vectors and mth taxonomic unit, respectively. Given a test set of k-mers (computed from reads), the distribution of the test set is modeled as

$$p(\mathbf{x}) = \sum_{m=1}^{M} p(\mathcal{C}_m)\ p(\mathbf{x}|\mathcal{C}_m), \tag{1}$$

where we denote probability for taxonomic unit m (or class weight) by $p(\mathcal{C}_m)$, satisfying $\sum_{m=1}^{M} p(\mathcal{C}_m) = 1$. Note that $\{p(\mathcal{C}_m)\}_{m=1}^{M}$ is the composition of taxonomic units in the given test set (reads). The inference task is to estimate $p(\mathcal{C}_m)$ as accurately as possible with a reasonable computational resource. Let us derive the mean vector

$$\mathbb{E}[\mathbf{x}] = \int \mathbf{x}\, p(\mathbf{x})\, d\mathbf{x} = \sum_{m=1}^{M} p(\mathcal{C}_m)\ \int \mathbf{x}\ p(\mathbf{x}|\mathcal{C}_m)\, d\mathbf{x}. \tag{2}$$

The mean $\mathbb{E}[\mathbf{x}]$ contains information about $p(\mathcal{C}_m)$ in this probabilistic formulation. In practice, the information summary is obtained by computing the sample mean from the complete set of reads available for a sample. Let us denote the sample mean of k-mers feature vectors of reads by $\boldsymbol{\mu} \in \mathbb{R}_+^{4^k}$ with the assumption that $\boldsymbol{\mu} \approx \mathbb{E}[\mathbf{x}]$. Several methods, such as Taxy [14], Quikr [11], and SEK [4] use the sample mean $\boldsymbol{\mu}$ directly as the main input to compute the composition $p(\mathcal{C}_m)$.

2.2 Segregation of read data by K-means (ARK)

For the above-described principle of information aggregation from the reads by the mean vector of k-mer counts, computation of the sample mean vector is straightforward. This consequently enables handling of a very large amount of reads with low computational cost. However, we hypothesize that the sample mean vector computed from the full set of reads is not sufficient in terms of information content to facilitate accurate estimation of $p(\mathcal{C}_m)$. Hence, we segregate the reads into several subsets and compute a sample mean vector separately for each subset, assuming that a set of sample mean vectors is more informative than a single mean vector. Note that in the case where the resulting read subsets were not in practice distinct from each other in terms of their k-mer counts, the subsequent composition estimate would effectively be identical to the estimate obtained with a single data summary described in section 2.1.

Let us partition the k-mers feature space $\mathbb{R}_+^{4^k}$ into Q non-overlapping regions $\mathcal{R}_q$ such that $\cup_{q=1}^{Q} \mathcal{R}_q = \mathbb{R}_+^{4^k}$ and $\forall q, r,\ q \neq r,\ \mathcal{R}_q \cap \mathcal{R}_r = \emptyset$. Such partitions can be formed by a standard K-means algorithm that typically uses a nearest neighbor classification rule based on square Euclidean distance measure. The non-overlapping regions $\mathcal{R}_q$ are called Voronoi regions. We define $P_q \triangleq \Pr(\mathbf{x} \in \mathcal{R}_q)$ satisfying $\sum_{q=1}^{Q} P_q = 1$. In practice, P_q is computed as

$$P_q = \frac{\text{number of feature vectors in } \mathcal{R}_\text{q}}{\text{total number of feature vectors}}. \tag{3}$$

It is reminded that the feature vectors are k-mers. The distribution of the full test set and subsets can be written as

$$\begin{aligned} &p(\mathbf{x}) = \textstyle\sum_{q=1}^{Q} P_q\ p(\mathbf{x}|\mathbf{x} \in \mathcal{R}_q), \\ &p(\mathbf{x}|\mathbf{x} \in \mathcal{R}_q) = \textstyle\sum_{m=1}^{M} p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)\ p(\mathbf{x}|\mathcal{C}_m, \mathbf{x} \in \mathcal{R}_q), \end{aligned} \tag{4}$$

where the first equation follows a standard mixture density framework. Now, if we can estimate $p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)$, then the final quantity of interest can be computed as

$$p(\mathcal{C}_m) = \sum_{q=1}^{Q} P_q\ p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q). \tag{5}$$

Let us now derive the mean vector for $\mathcal{R}_q$, which is a conditional mean vector

$$\begin{aligned} &\mathbb{E}[\mathbf{x}|\mathbf{x} \in \mathcal{R}_q] \\ &= \textstyle\int \mathbf{x}\ p(\mathbf{x}|\mathbf{x} \in \mathcal{R}_q)\, d\mathbf{x} \\ &= \textstyle\sum_{m=1}^{M} p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)\ \int \mathbf{x}\ p(\mathbf{x}|\mathcal{C}_m, \mathbf{x} \in \mathcal{R}_q)\ d\mathbf{x}. \end{aligned} \tag{6}$$

The mean $\mathbb{E}[\mathbf{x}|\mathbf{x} \in \mathcal{R}_q]$ contains information about $p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)$. In practice we use the sample mean denoted by $\boldsymbol{\mu}_q$ with the assumption that $\boldsymbol{\mu}_q \approx \mathbb{E}[\mathbf{x}|\mathbf{x} \in \mathcal{R}_q]$. Comparing (2) and (6), for the qth Voronoi region $\mathcal{R}_q$ we can estimate composition $p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)$ by using an appropriate composition estimation method, such as Taxy, Quikr or SEK.

2.3 Algorithms

The ARK algorithm can be implemented by following steps.

a. Divide the full test dataset of k-mers into Q subsets. The region $\mathcal{R}_q$ corresponds to the qth subset.
b. For the qth subset, compute P_q and the sample mean $\boldsymbol{\mu}_q$.
c. For the qth subset, apply a composition estimation method that uses the input $\boldsymbol{\mu}_q$; estimate $p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)$.
d. Estimate $p(\mathcal{C}_m)$ by $p(\mathcal{C}_m) = \sum_{q=1}^{Q} P_q \; p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)$.

The ARK method is described using a flow-chart in Figure 1. The flow-chart shows the main components of the overall system and the associated off-line and on-line computations. The crucial computational/statistical challenges related to the ARK algorithm outlined above are as follows:

1. What is an appropriate number of subsets Q?
2. How should one form the subsets $\mathcal{R}_q$?

The above points are inherent to any subset forming algorithm, and more generally to any clustering algorithm. Furthermore, finding optimal regions (or clusters) requires alternative optimization techniques. Given a pre-defined Q, typically a K-means algorithm performs two alternating optimization steps. These are: (1) given a set of representation vectors $\{\boldsymbol{\mu}_q\}_{q=1}^{Q}$ (also called code vectors) form new clusters $\{\mathcal{R}_q\}_{q=1}^{Q}$ by a nearest neighbor rule (or form new subsets from the full dataset), (2) find the set of cluster representation vectors given the assignment of data into clusters. The optimal representation vector is the mean vector if squared Euclidean distance is used for the nearest neighbor rule. The K-means algorithm starts with an initialization of set of representation vectors and runs alternating optimization until convergence in the sense that the average squared Euclidean distance is no longer reduced. In the present paper we perform the clustering using a popular vector quantization method called the Linde-Buzo-Gray (LBG) algorithm [13] (or source coding literature). There are several variants of the LBG available. In one variant, the algorithm starts with $Q = 1$ and then slowly splits the dense and high probability clusters to end up with a high Q, such that it does not deviate significantly from exponentially decaying bit rate versus coding distortion (rate-distortion) curve.

In ARK, we use the following two strategies to solve the two challenges listed above.

1. Optimal/deterministic strategy: Start with $Q = 1$, which corresponds to the previous approach with a single mean vector as the data summary. Then set $Q = 2$ for LBG algorithm that uses square Euclidean distance as the distortion measure; the LBG algorithm minimizes mean of square Euclidean distance (also called mean square error). Initialization is done by a standard split approach where the mean vector is perturbed. Using $Q = 2$, $\{\mathcal{R}_q\}_{q=1}^{2}$ is formed and we estimate $p(\mathcal{C}_m)$. Subsequently, Q is increased by one until a convergence criterion is met. For $Q \geq 3$, we always split the highest ranking cluster into two subclusters and use the LBG algorithm to find the optimal clusters. The number of clusters Q is no longer increased if the estimated values of $p(\mathcal{C}_m)$ differ negligibly for Q and $(Q-1)$.

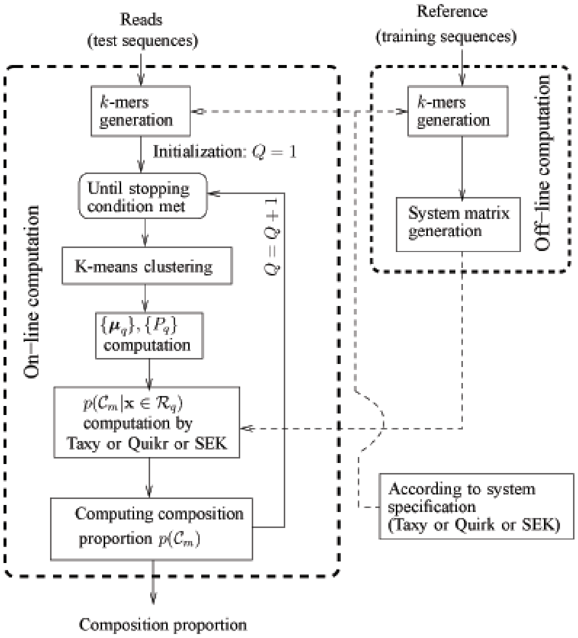
Reads
(test sequences)
Reference
(training sequences)
k-mers generation
Initialization: $Q = 1$
Until stopping condition met
K-means clustering
$Q = Q + 1$
$\{\boldsymbol{\mu}_q\}, \{P_q\}$ computation
$p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)$ computation by Taxy or Quikr or SEK
Computing composition proportion $p(\mathcal{C}_m)$
On−line computation
k-mers generation
System matrix generation
Off−line computation
According to system specification (Taxy or Quirk or SEK)
Composition proportion

Fig. 1. A flow-chart of the ARK method.

In practice, the stopping condition we use is that the variational distance between $p(\mathcal{C}_m)|_Q$ and $p(\mathcal{C}_m)|_{(Q-1)}$ is less than a predetermined threshold. This condition can be written as $\sum_{m=1}^{M} \text{abs}\left(p(\mathcal{C}_m)|_Q - p(\mathcal{C}_m)|_{(Q-1)}\right) < \eta$, with a user defined choice of the threshold η. This strategy is typically found to provide consistent performance improvement in the sense of estimating $p(\mathcal{C}_m)$ with the increase in Q by the step of one, but without absolute guarantee as the target optimization strategy minimizes mean square error. Furthermore, we allow an increment in the number of clusters up to a pre-defined maximum limit Q_{max}. Typically Q_{max} is chosen an integer power of two.

2. Non-optimal/random strategy: For very large test sets, we use a pre-determined Q and a random choice of the Q representation vectors. Then the full test set is divided into Q subsets by a nearest neighbor rule and we compute the set of Q mean vectors $\{\boldsymbol{\mu}_q\}$, and cluster probabilities $\{P_q\}$. Even though this non-optimal strategy does not use an alternating optimization (such as LBG algorithm) to form optimal clusters, it divides the full test set into sub-sets, resulting in a set of Q localized mean vectors across the full test set.

Finally we mention that the use of K-means is fully motivated by its simplicity and computational ease. Use of statistical K-means in the form of expectation-maximization based mixture modeling (for example, Gaussian mixture model) could have been implemented, but requires more computation to handle a large dataset of reads.

2.4 Synthetic data generation for method evaluation

To evaluate the performance of the ARK method, we conducted experiments for simulated data as described below. For these, and all computations reported in the remainder of the paper, we used

Matlab version R2013b (with some instances of C code), on a desktop workstation with an Intel Core i7 4930K processor and 64Gb of RAM.

2.4.1 Test datasets (Reads): We simulated 180 16S rRNA gene datasets using the RDP training set 7 and the Grinder read simulator [1] targeting the V1-V2 and V3-V5 variable regions with read lengths fixed at 250 bp or normally distributed with a mean of 450 bp and variance 50 bp. Read depths were chosen to be either 10K, 100K or 250K, while three different read distributions were used: power law, uniform, and linear. Diversity was set at either 50, 100, or 500 organisms and chimera percentages were set to 5% or 35%. The Balzer model [2] was chosen for homopolymer errors, and copy bias was included while length bias was excluded.

2.4.2 Training datasets (Reference): In ARK experiments we used Quikr [11] and SEK [4] to estimate $p(\mathcal{C}_m|\mathbf{x} \in \mathcal{R}_q)$. The RDP training set 7 was used as the base reference database for both Quikr and SEK.

2.5 Real biological data

To further evaluate ARK, we also utilized 28 Illumina MiSeq 16S rRNA gene human body-site associated samples, plus one negative control sample. The real data consist of a total of over 5.7M reads distributed over three variable regions (V1-V2, V3-V4, and V3-V5) as well as two body sites (vaginal and feces).

For each of these samples DNA was extracted using the FastDNA SPIN Kit for Soil with a FastPrep machine (MP Biomedicals) following the manufacturers protocol. 16S rRNA gene amplicons were generated from the DNA extractions using the primer combinations listed in Section 5 in the Supplementary Data. The Q5 High-fidelity polymerase kit (New England Biolabs) was used to amplify the 16S rRNA genes, and PCR conditions were as follows: 98°C for 2 minutes, followed by 20 cycles of 98°C for 30 seconds, 50°C for 30 seconds and 72°C for 1 minute 30 seconds, followed by a final extension step at 72°C for 5 minutes. Following PCR, the amplicons were then purified using the Wizard SV Gel and PCR Clean-Up kit (Promega, UK). Sequencing of 16S rRNA gene amplicons was carried out by Illumina Inc. (Little Chesterford, UK) using a MiSeq instrument run for 2 x 250 (V1-V2), 300 + 200 (V3-V4) and 400 + 200 (V3-V5) cycles.

After trimming 20bp of primer off each read, the sequences were trimmed from the right until all bases had a quality score greater than 27. This reduced the total number of reads to approximately 4M, and reduced the mean read length from 315 bp to 257 bp. We then utilized all resulting unpaired reads (both forward and reverse) including any duplicate sequences.

3 RESULTS

3.1 Performance measure and relevant methods

As a quantitative performance measure, we use variational distance (VD) to compare between known proportions of taxonomic units $\mathbf{p} = [p(\mathcal{C}_1), p(\mathcal{C}_2), \dots, p(\mathcal{C}_M)]^t$ and the estimated proportions $\hat{\mathbf{p}} = [\hat{p}(\mathcal{C}_1), \hat{p}(\mathcal{C}_2), \dots, \hat{p}(\mathcal{C}_M)]^t$. The VD is defined as

$$\mathrm{VD} = 0.5 \times \|\mathbf{p} - \hat{\mathbf{p}}\|_1 \in [0, 1].$$

A low VD indicates more satisfactory performance.

For ARK, we used both SEK and Quikr as the underlying estimation methods applied to each cluster. These recent methods were chosen as appropriate representatives of fast and accurate sparse signal processing approaches. A k-mer size of $k = 6$ was used for both Quikr and SEK.

As part of the SEK pipeline, sequences in a given database are split into subsequences. We selected from the 10,046 sequences in the RDP training set 7 all sequences longer than 700 bp in length, and then split the sequences into subsequences of length 400 bp with 100 bp of overlap. This corresponds to setting $L_w = 400$ and $L_p = 100$ as specified in [4]. We used the SEK algorithm $\mathrm{OMP}^{+,1}_{\mathrm{sek}}$ with parameters as in [4].

3.2 Results for Simulated Data

3.2.1 Effect of increasing number of clusters: We first investigate how an increase in the number of clusters Q affects the composition reconstruction fidelity and algorithm execution time for the simulated data. Only the non-optimal/random strategy of K-means clustering was utilized as we found that the performance improvement for optimal/deterministic strategy was insignificant given the resulting increase in execution time (results not shown). Averaging the VD error at the genus level over all 180 simulated experiments, it was found that combining ARK with both SEK and Quikr resulted in a power law kind of decay of VD error as a function of the number of clusters. The left panel of Figure 2 demonstrates this. ARK causes a substantial increase in reconstruction fidelity which can be seen since using ARK SEK or ARK Quikr with one cluster is equivalent to running SEK or Quikr with no modification.

Since the underlying algorithm (SEK or Quikr) must be executed on each cluster formed by the K-means clustering, we expect the total algorithm execution time to increase by a factor equal to the number of chosen clusters. As seen in the right panel of Figure 2, both algorithms experience an increase in execution time roughly proportional to the number of clusters.

3.2.2 Fixed number of clusters: Given the decrease of VD as a function of the number of clusters seen in the Section 3.2.1, we also fixed the number of clusters Q to 75 to compare the performance of the underlying algorithms with and without ARK. There was a significant decrease in the VD error (as seen in the left panel of Figure 3) at the cost of an increase in execution time (the right panel of Figure 3). However, given the speed of both Quikr and SEK, we expect the addition of ARK will not result in prohibitively long execution times. Indeed, as seen in section 3.3, on real biological data both ARK Quikr and ARK SEK are still several hours faster than RDP's NBC even when using 75 clusters.

3.3 Real Biological Data

We used ARK combined with SEK and Quikr to analyze the biological data detailed in section 2.5 and compared these results to those obtained from the Ribosomal Database Project's Naïve Bayesian Classifier (RDP's NBC) [21]. All methods used RDP's training set 7 as the underlying training database. The random K-means clustering was used for the ARK method, and the number of clusters Q was set to 75. Figure 4 demonstrates the total execution time of each method. While ARK does increase the execution time of Quikr and SEK, the total execution time is still significantly less

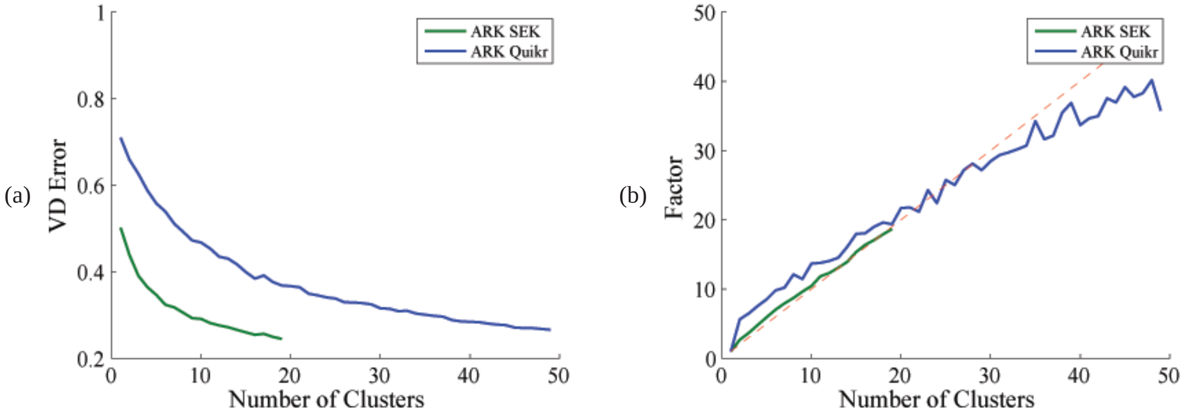

Fig. 2. Results for the random K-means clustering on the simulated data. (a) The left panel: Mean VD error at the genus level as a function of the number of clusters. Note the improvement that ARK contributes to each method. (b) The right panel: Mean execution time increase (factor given in comparison to running SEK or Quikr in the absence of ARK) as a function of number of clusters. The dashed line represents a line with slope 1.

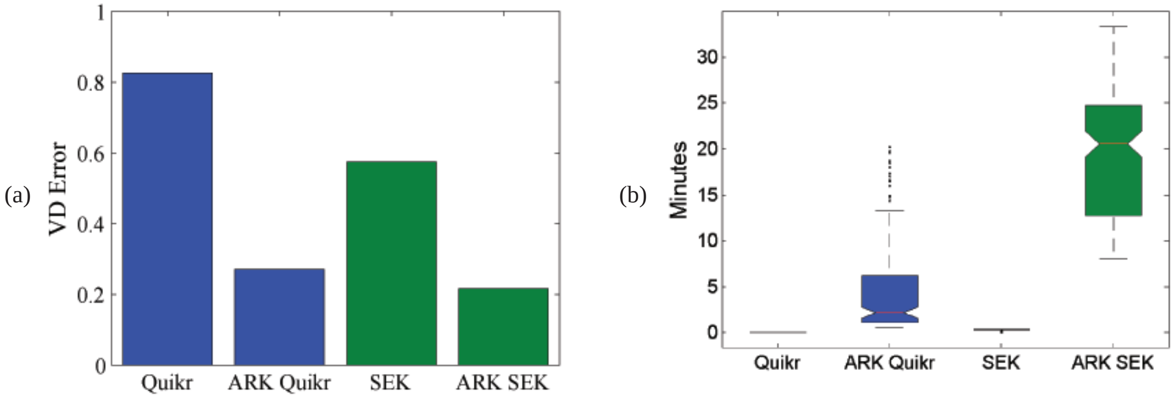

Fig. 3. Comparison of the underlying algorithms with and without ARK. Results are for the random K-means clustering on the simulated data when fixing the number of clusters to 75. (a) The left panel: Mean VD error at the genus level. (b) The right panel: Boxplot of the individual simulated sample execution times. Mean execution times for Quikr and ARK Quikr were 1.75 seconds and 4.71 minutes, while for SEK and ARK SEK they were 21.26 seconds and 19.21 minutes respectively.

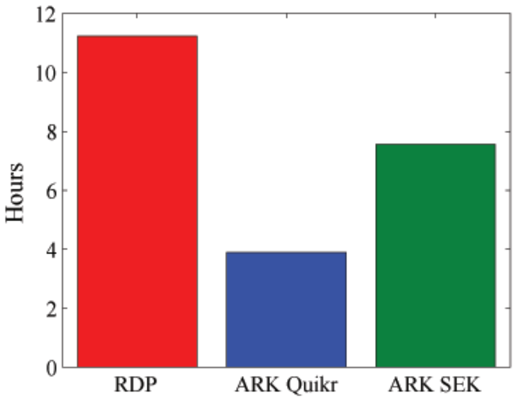

Fig. 4. Total execution time for each method on the 28 samples of real biological data.

than that of RDP's NBC. Note that all datasets here are not de-duplicated. Execution time of RDP's NBC can be accelerated by de-duplicating the data before classifying. However, this requires additional computational time to find duplicate sequences, and since we are directly comparing classification methods here (not computational shortcuts) we use the same non-de-duplicated data for all methods.

To compare the results of each method, we compared PCoA (also known as classical multidimensional scaling) plots by employing the Jensen-Shannon divergence on each of the reconstructions. The points represent individual samples, and the color/shape denote the associated metadata. Each of the methods produced similar PCoA plots. Figure 5 compares the results when using RDP's NBC (the left panel) and ARK SEK (the right panel) when the sample body site is labeled. Note the similar clusterings.

As shown in Figure 6, while ARK Quikr gave a somewhat similar PCoA plot with regard to body site (the left panel), clustering by variable region (the right panel) was also observed. This is most likely due to the fact that different variable regions have different k-mer distributions. ARK Quikr can detect this as it analyzes each sample in its entirety, as opposed to the read-by-read nature of RDP's NBC. This is corroborated by the fact that when using the

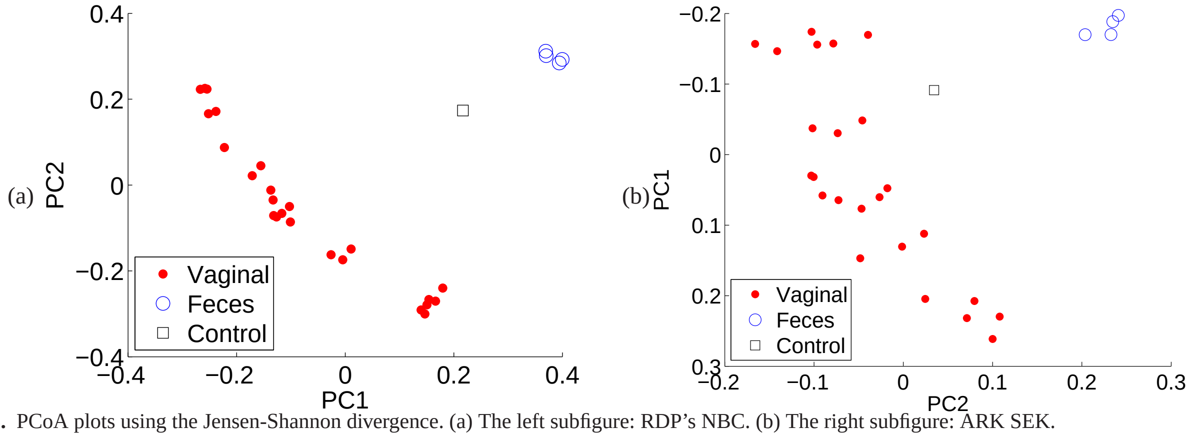

Fig. 5. PCoA plots using the Jensen-Shannon divergence. (a) The left subfigure: RDP's NBC. (b) The right subfigure: ARK SEK.

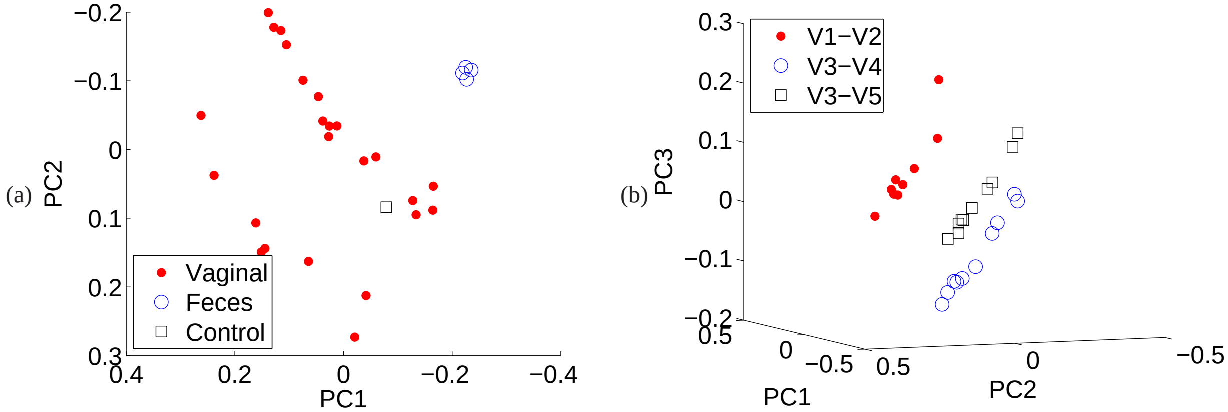

Fig. 6. ARK Quikr PCoA plots (using the Jensen-Shannon divergence) on the real biological data. (a) The left panel: Labeling by body site. (b) The right panel: Labeling by variable region. Note the clustering for both labelings.

Jenson-Shannon divergence directly on the 6-mer counts, similar grouping was observed by variable region (results not shown).

4 DISCUSSION AND CONCLUSION

The addition of a data processing step based on clustering the read information prior to community composition estimation is akin to the generic divide-and-conquer principle used judiciously in the machine learning field. In terms of information content of the read data, the individual means of the k-mer frequencies can collectively provide a better summary than the single mean vector used in the previous approaches, when sufficient heterogeneity is present among the sequences. Our experiments demonstrate this effect by a substantial increase in the accuracy of the resulting estimates. Moreover, the clustering employed by ARK is found to be robust in the sense that it does not lead to lower accuracies, even if a suboptimal number of clusters and clustering strategy were used. We found that the improvement in reconstruction accuracy was obtained at the cost of a moderate increase in execution time for the studied methods.

We note that under the clustering algorithm employed by ARK, no quantitative claims can be made concerning the global optimality of the resulting clusters or on consistent improvement in performance. Also, there is no absolute guarantee that the estimation of $p(\mathcal{C}_m)$ is bound to improve monotonically with an increase in Q. Thus, in an individual experiment, it is possible to encounter occasional decreases in performance. However, our results suggest that a larger number of clusters Q will tend to perform reasonably better than a much smaller value of Q, provided that the resulting cluster sizes are not too small to yield very noisy estimates of the mean vector.

While this study has focused on 16S rRNA gene sequencing based data, there is no theoretical limitation in applying this technique also to whole-genome shotgun (WGS) metagenomics. Indeed, ARK can readily be combined with existing WGS k-mer feature vector metagenomics reconstruction techniques (such as WGSQuikr [12]). Thus, we aim at investigating the versatility of this approach as complementary to other WGS metagenomics analysis methods in the future.

ACKNOWLEDGEMENT

Funding: This work was supported by the Swedish Research Council Linnaeus Centre ACCESS (S.C.), ERC grant 239784 (J.C.), the Academy of Finland Center of Excellence COIN (J.C.), the Academy of Finland (M.V.), the Scottish Government's Rural and Environment Science and Analytical Services Division (RESAS)

(A.W.W) and the UK MRC/DFID grant G1002369 (SCF). The authors wish to thank Paul Scott (Wellcome Trust Sanger Institute) for generating the 16S rRNA gene amplicons used in the real biological analyses. Real biological data from vaginal sites was collected in Mwanza, Tanzania from the Mwanza Intervention Trials Unit, National Institute of Medical Research.

4.0.1 Conflict of interest statement. None declared.

REFERENCES

[1]F. E. Angly, D. Willner, F. Rohwer, P. Hugenholtz, and G. W. Tyson. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic Acids Research*, 40(12):e94, 2012.

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

[3]Y. Cai and Y. Sun. Esprit-tree: hierarchical clustering analysis of millions of 16s rrna pyrosequences in quasilinear computational time. *Nucleic Acids Research*, 39(14):e95, 2011.

[4]S. Chatterjee, D. Koslicki, S. Dong, N. Innocenti, L. Cheng, Y. Lan, M. Vehkaperä, M. Skoglund, L.K. Rasmussen, E. Aurell, and J. Corander. Sek: Sparsity exploiting k-mer-based estimation of bacterial community composition. *Bioinformatics*, 30(17):2423–31, 2014.

[5]S. Chatterjee and T.V. Sreenivas. Optimum switched split vector quantization of lsf parameters. *Signal Processing*, 88(6):1528–1538, 2008.

[6]L. Cheng, A.W. Walker, and J. Corander. Bayesian estimation of bacterial community composition from 454 sequencing data. *Nucleic Acids Research*, 2012.

[7]J Dröge, I Gregor, and AC McHardy. Taxator-tk: Precise taxonomic assignment of metagenomes by fast approximation of evolutionary neighborhoods. *Bioinformatics*, 2014.

[8]R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2010.

[9]R. C. Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.

[10]D.H. Huson, A.F. Auch, J. Qi, and S.C. Schuster. Megan analysis of metagenomic data. *Genome Res.*, 17(3):377–386, 2007.

[11]D. Koslicki, S. Foucart, and G. Rosen. Quikr: a method for rapid reconstruction of bacterial communities via compressive sensing. *Bioinformatics*, 29(17):2096–2102, 2013.

[12]D Koslicki, S Foucart, and G Rosen. WGSQuikr: Fast Whole-Genome Shotgun Metagenomic Classification. *PloS one*, 9(3):e91784, 2014.

[13]Y. Linde, A. Buzo, and R.M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, 1980.

[14]P. Meinicke, K.P. Aßhauer, and T. Lingner. Mixture models for analysis of the taxonomic composition of metagenomes. *Bioinformatics*, 27(12):1618–1624, 2011.

[15]S. Mitra, M. Stärk, and D.H. Huson. Analysis of 16s rrna environmental sequences using megan. *BMC Genomics*, 2011.

[16]S.H. Ong, V.U. Kukkillaya, A. Wilm, C. Lay, E.X.P. Ho, L. Low, M.L. Hibberd, and N. Nagarajan. Species identification and profiling of complex microbial communities using shotgun illumina sequencing of 16s rrna amplicon sequences. *PLoS One*, 8(4):e60811, 2013.

[17]H.H. Otu and K. Sayood. A divide-and-conquer approach to fragment assembly. *Bioinformatics*, 19(1):22–29, 2003.

[18]G. Rosen, E. Garbarine, D. Caseiro, R. Polikar, and B. Sokhansanj. Metagenome Fragment Classification Using k-Mer Frequency Profiles. *Advances in Bioinformatics*, 2008, 2008.

[19]G. Rosen, E. Reichenberger, and A. Rosenfeld. NBC: the Naïve Bayes Classification tool webserver for taxonomic classification of metagenomic reads. *Bioinformatics*, 27(1):127, 2011.

[20]C. von Mering, P. Hugenholtz, J. Raes, S. G. Tringe, T. Doerks, L. J. Jensen, N. Ward, and P. Bork. Quantitative phylogenetic assessment of microbial communities in diverse environments. *Science*, 315(5815):1126–1130, 2007.

[21]Q. Wang, G.M. Garrity, J.M. Tiedje, and J.R Cole. Naïve bayesian classifier for rapid assignment of rrna sequences into the new bacterial taxonomy. *Appl. Environ. Microbiol.*, 73(16):5261–5267, 2007.

BIOINFORMATICS

Vol. 00 no. 00 2015
Pages 1–3

Supplementary Data for ARK: Aggregation of Reads by K-means for Estimation of Bacterial Community Composition

Saikat Chatterjee [1]*, Damon Shahrivar [1], David Koslicki [2], Alan W. Walker [3], Suzanna C. Francis[4], Louise J. Fraser[5], Mikko Vehkaperä [6], Yueheng Lan [7], Jukka Corander [8]

[1]Dept of Communication Theory, KTH Royal Institute of Technology, Sweden
[2]Dept of Mathematics, Oregon State University, Corvallis, USA
[3]Microbiology Group, Rowett Institute of Nutrition and Health, University of Aberdeen, UK
[4]MRC Tropical Epidemiology Group, London School of Hygiene and Tropical Medicine, London, UK
[5]Illumina Cambridge Ltd., Chesterford Research Park, Essex, UK
[6]Dept of Signal Processing, Aalto University, Finland
[7]Dept of Physics, Tsinghua University, Beijing, China
[8]Dept of Mathematics and Statistics, University of Helsinki, Finland

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

1 REASON FOR SUPPLEMENTARY DATA

This supplementary material is included to address five major points:

1. To compare ARK with the best performing bacterial community composition method to date, called BEBaC [3]. BEBaC employs a Bayesian estimation clustering framework alongwith a stochastic search and sequence alignment.
2. To investigate the important question of finding the number of regions Q in ARK.
3. To independently verify ARK in two different geographic regions ((1) Sweden and Finland, and (2) USA) and also using different datasets.
4. To detail genera-level reconstructions of ARK SEK, ARK Quikr, and RDP's NBC.
5. To detail the primers used to obtain the data from section 2.5.

2 COMPARING ARK TO BEBAC AND FINDING THE NUMBER OF REGIONS Q

2.1 Mock communities data

BEBaC is very accurate, but at the expense of substantial computational cost: a running time of several days in a computing-cluster environment may be required for large read sets. Therefore, to compare ARK with BEBaC, we used a dataset for which the result of BEBaC is readily available. Furthermore, finding Q (the second point above) is also computationally intensive, so the reasonable size of the dataset helped in this regard.

For our experiments on real biological data, we used the mock microbial communities database developed in [4]. The database is called even composition Mock Communities (eMC) for chimeric sequence detection where the bacterial species included are known in advance. Three regions (V1-V3, V3-V5, V6-V9) of the 16S rRNA gene of the eMC were sequenced using 454 sequencing technology in four different sequencing centers. In our experiments we focused on the V3-V5 region datasets, since these have been previously used for the evaluation of the BEBaC method (see Experiment 2 of [3]) and SEK, Quikr and Taxy methods (see Experiment in Section 3.2.2 of [2]).

2.1.1 Test dataset (Reads): Our basic test dataset used under a variety of different *in silico* experimental conditions is the one used in Experiment 2 of BEBaC [3] and the Experiment in Section 3.2.2 of SEK [2]. The test dataset consists of 91,240 short reads from 21 different species. The length of reads has a range between 450-550 bp and the bacterial community composition is known at the species level by using the following computation performed in [3]. Each individual sequence of the 91,240 read sequences was aligned (local alignment) to all the reference sequences of reference database described in the section 2.1.2 and then each read sequence is labeled by the species of the highest scoring reference sequence, followed by computation of the community composition referred to as ground truth.

2.1.2 Training datasets (Reference): We used a database generated from the eMC database [4]. The database consists of the same $M = 21$ species present among the reads described in section 2.1.1. The details of the reference database can be found in Experiment 2 of BEBaC [3]. The database was also used for training SEK, Quikr and Taxy in the experiment described in Section 3.2.2 of [2]. The

[*] To whom correspondence should be addressed. Email: sach@kth.se

database consists of 113 reference sequences for a total of 21 bacterial species, such that each reference sequence represents a distinct 16S rRNA gene. Thus there is a varying number of reference sequences for each of the considered species. Each reference sequence has an approximate length of 1500 bp, and for each species, the corresponding reference sequences are concatenated to a single sequence. The final reference database consists of 21 sequences where each sequence has an approximate length of 5000 bp.

3 RESULTS

3.1 Computational resources and reproducible codes

We used standard Matlab software. For hardware, we used a Dell Latitude E6400 laptop computer with a 3 GHz processor and 8 GB memory. We also used the **cvx** [1] convex optimization toolbox. The reproducible Matlab codes are available at http://www.ee.kth.se/ctsoftware.

3.2 Relevant methods

For ARK, we use SEK, Taxy [5] and Quikr as the estimation methods employed to each cluster. The choice of SEK is due to its recent development and good performance. The use of Taxy and Quikr is due to fact that they also use the first order moment of the k-mer frequencies of reads. We use the greedy algorithm, called $\mathrm{OMP}_{\mathrm{sek}}^{+,1}$ to realize SEK. We also use the parameters for $\mathrm{OMP}_{\mathrm{sek}}^{+,1}$ and parameters for k-mers training and testing setups identical to that used in [2] so that fair comparison can be pursued. Finally we compared the performance of ARK-SEK with the ground truth and BEBaC.

3.3 Results for Mock Communities data

Using mock communities data, we carried out experiments where the community composition problem was addressed at the species level.

3.3.1 Value of k for k-mers: Throughout all experiments, we used $k = 4$ for reads and reference. Therefore the k-mer feature vectors have a dimension of 256.

3.3.2 k-mers from test dataset: In the test dataset, described in section 2.1.1, the shortest read is 450 bp long. Therefore, for each read sequence, we used first 450 bp sequence to compute k-mers. Using $k = 4$, the generation of k-mers feature vectors from reads (testset) took 21 minutes of execution time. Note that the k-mers test dataset is same as that used in Section 3.2.2 of [2].

3.3.3 Results: We first investigated the convergence of ARK and how the performance of ARK behaves with increasing cluster number. We used $\eta = 0.0005$. The performance of ARK using three estimation methods is shown in Figure 1. We note that the gross performance trend typically improves with the increase in clusters and then saturates. ARK-SEK saturates at 24 clusters with the VD of 0.0103. The ARK-SEK performance is much better than ARK-Taxy and ARK-Quikr. The scope of improvement of SEK via ARK-SEK is found to be limited. In contrast, figure 2 shows that Quikr is found to improve significantly via ARK-Quikr (after convergence).

The convergence time of ARK-SEK was 763 seconds ($\sim$12 minutes). Note that K-means clustering is common for all ARK methods either using SEK, Taxy and Quikr. As ARK-Quikr converged at 17 clusters and took 18 seconds in total, it is clear that the clustering algorithm does not require heavy computational resource. Typically K-means clustering is a simple algorithm and hence can be used for a very large size read dataset without much demand in computational resource.

Finally we compare SEK, ARK-SEK and BEBaC against the ground truth in Figure 3. BEBaC results are highly accurate with VD = 0.0038, but comes with the requirement of a computation time in the order of more than a day. But ARK-SEK provides a reasonable performance at the expense of total online computation time of $21 + 12 = 33$ minutes. While BEBaC is assumed to incapable of handling a very large dataset of reads, ARK-SEK can be used for this scenario.

4 GENUS-LEVEL RECONSTRUCTIONS

To provide more detailed insight into how ARK Quikr and ARK SEK perform in comparison to RDP's NBC, we include here genus level reconstructions of a few real data sets (detailed in section 2.5 in the main text). We select three sample spanning three variable regions (V1-V2, V3-V5, and V3-V4) and two body sites (vaginal and feces). Since each method reconstructs a large number of very low abundance organisms, for simplicity of visualization, we restricted our attention to those genera which appear at an abundance level of $\geq 5\%$ in any of the three methods. Figure 4 gives a bar chart of each method's genera-level reconstruction on the three data sets. There was a general trend of the ARK SEK reconstruction agreeing more similarly to the RDP NBC results, as was observed in each of the samples contained in section 2.5 in the main text.

5 PRIMER DETAILS

For the samples detailed in section 2.5 in the main text, 16S rRNA gene amplicons were generated from the DNA extractions using the following primer combinations, where the letters in italics show the region targeting the 16S rRNA gene, non-italicized letters show Illumina adapter, primer pad and linker sequences and "nnnnnnnnnnnn" indicates where unique 12-base Golay barcode sequences were incorporated for each sample: V1-V2 = 27f (AATGATACGGCGACCACCGAGATCTACACTATGGTAATTCC*AGMGTTYGATYMTGGCTCAG*) and 338r (CAAGCAGAAGACGGCATACGAGATnnnnnnnnnnnnAGTCAGTCAGAA *GCTGCCTCCCGTAGGAGT*), V3-V4 = 338f (AATGATACGGCGACCACCGAGATCTACACTATGGTAATTGT *ACTCCTACGGGAGGCAGCAG*) and 806r (CAAGCAGAAGACGGCATACGAGATnnnnnnnnnnnnAGTCAGTCAGCC *GGACTACHVGGGTWTCTAAT*), V3-V5 = 338f (sequence as shown above) and 926r (CAAGCAGAAGACGGCATACGAGATnnnnnnnnnnnnAGTCAGTCAGCC *CCGTCAATTYMTTTRAGT*).

REFERENCES

[1] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

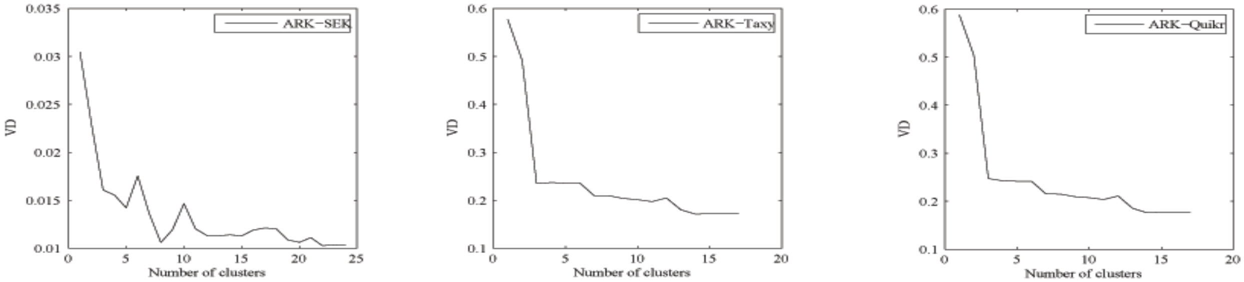

Fig. 1. For mock communities data: Performance of ARK with respect to number of clusters. The convergence time for ARK-SEK, ARK-Taxy and ARK-Quikr were 763 seconds, 61 seconds and 18 seconds respectively. Note that the ARK-Taxy and ARK-Quikr show significant improvement in performance by the use of clustering.

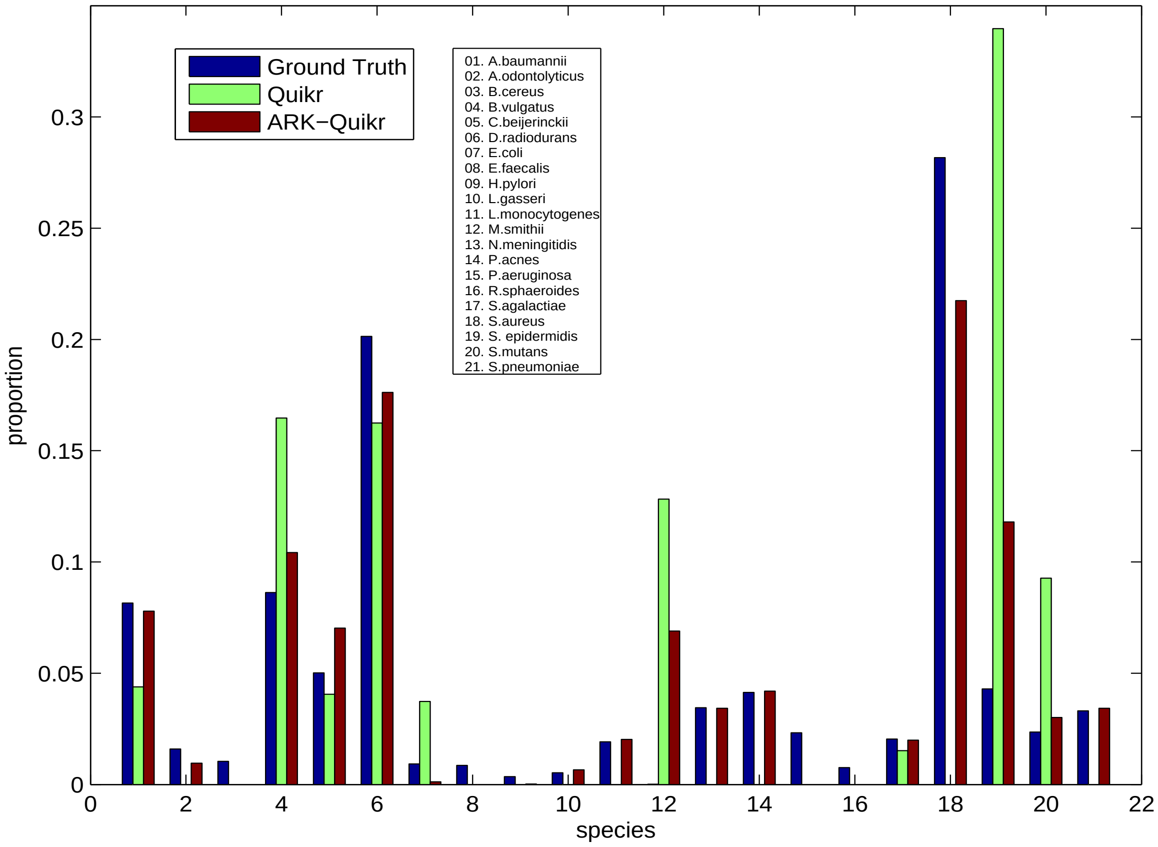

Fig. 2. Performance improvement of ARK-Quikr vis-a-vis Quikr and ground truth. Quikr and ARK-Quikr produced VD of 0.5882 and 0.1770 respectively.

[2]S. Chatterjee, D. Koslicki, S. Dong, N. Innocenti, L. Cheng, Y. Lan, M. Vehkaperä, M. Skoglund, L.K. Rasmussen, E. Aurell, and J. Corander. Sek: Sparsity exploiting k-mer-based estimation of bacterial community composition. *Bioinformatics*, 30(17):2423–31, 2014.

[3]L. Cheng, A.W. Walker, and J. Corander. Bayesian estimation of bacterial community composition from 454 sequencing data. *Nucleic Acids Research*, 2012.

[4]B.J. Haas, D. Gevers, A.M. Earl, M. Feldgarden, D.V. Ward, G. Giannoukos, D. Ciulla, D. Tabbaa, S.K. Highlander, E. Sodergren, B. Methe, T.Z. DeSantis, Human Microbiome Consortium, J.F. Petrosino, R. Knight, and B.W. Birren. Chimeric 16s rrna sequence formation and detection in sanger and 454-pyrosequenced pcr amplicons. *Genome Res.*, 21(3):494–504, 2011.

[5]P. Meinicke, K.P. Aßhauer, and T. Lingner. Mixture models for analysis of the taxonomic composition of metagenomes. *Bioinformatics*, 27(12):1618–1624, 2011.

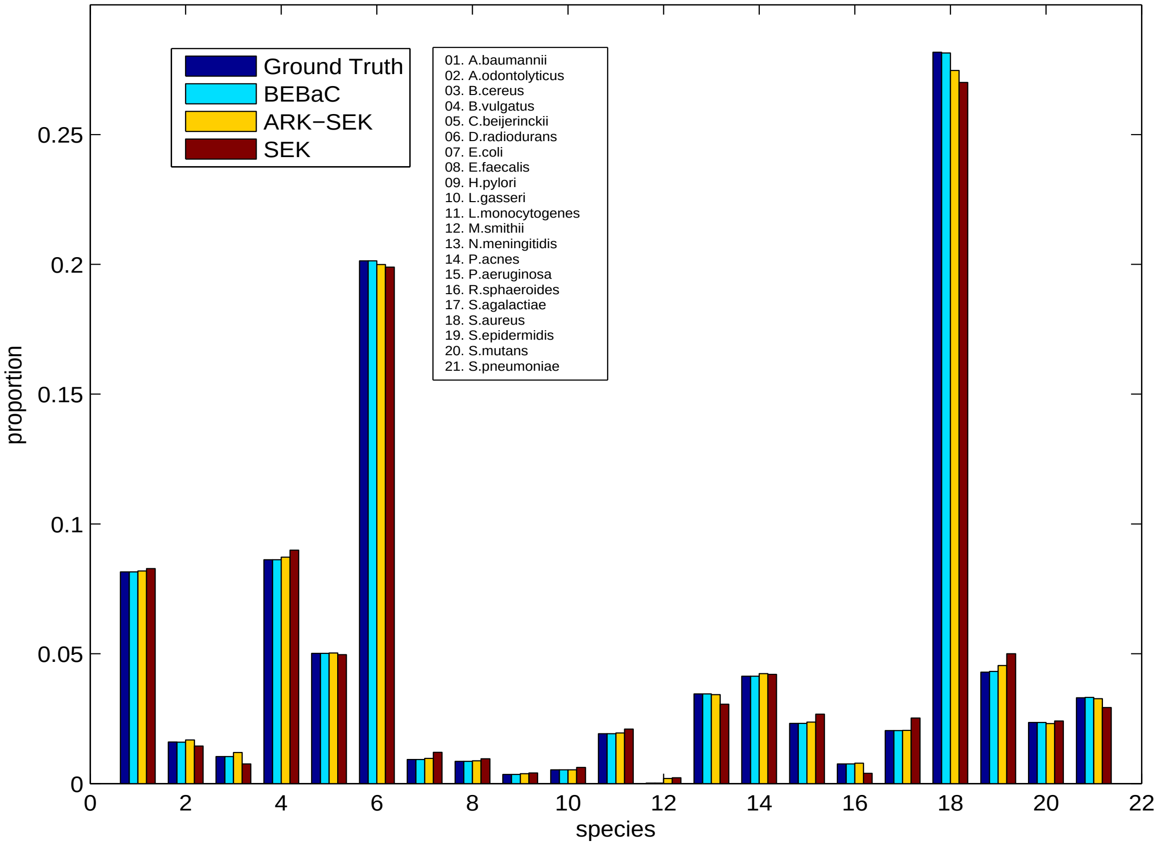

Fig. 3. Comparison of ARK-SEK, SEK and BEBaC against ground truth. BEBaC, ARK-SEK and SEK provide VD performance 0.0038, 0.0103 and 0.0305 respectively.

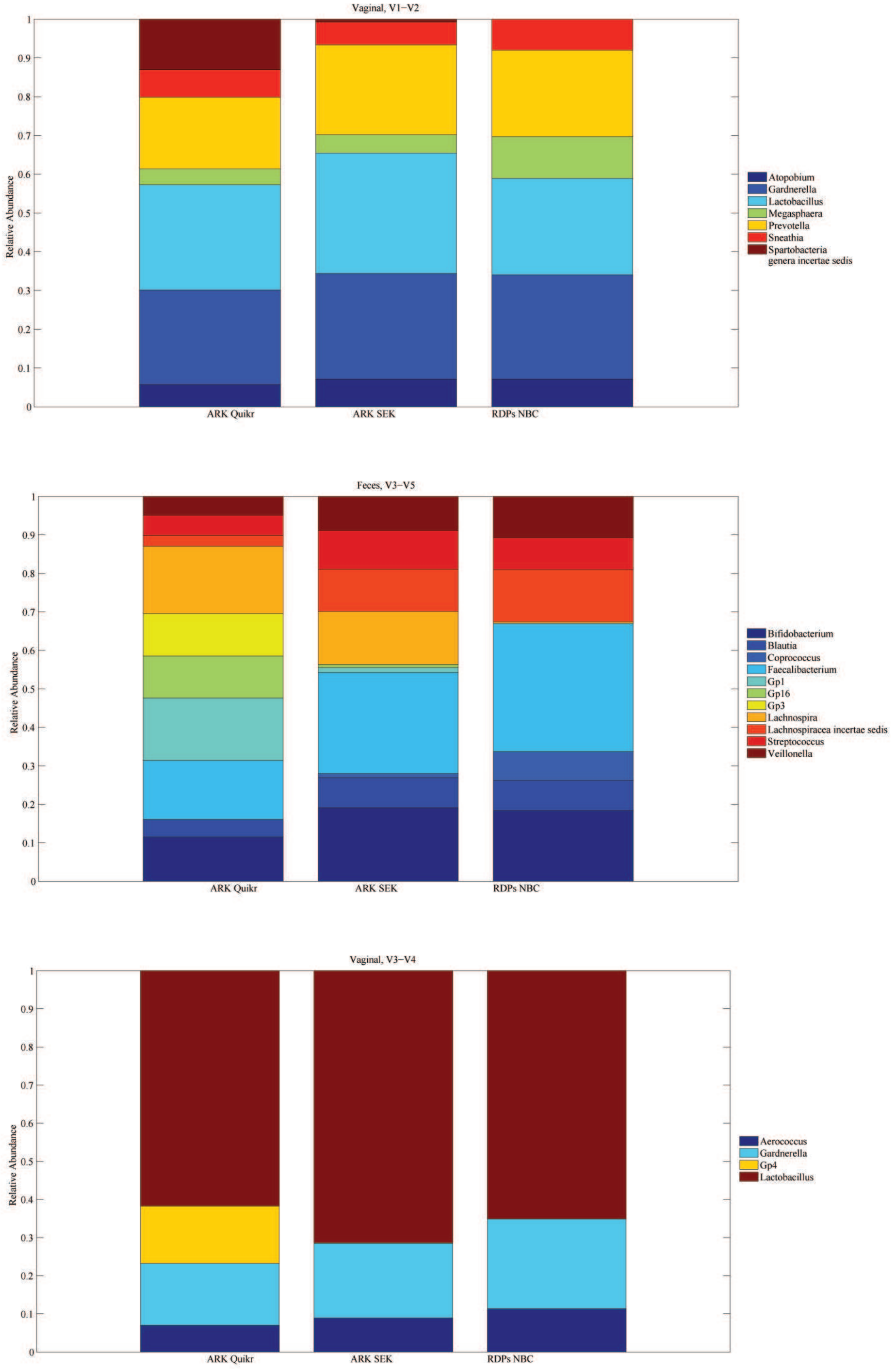

Fig. 4. Genus level reconstructions of ARK Quikr, ARK SEK, and RDP’s NBC on three of the real biological data sets from section 2.5 of the main text. Genera shown are those that have a proportional abundance of greater than or equal to 5% for at least one of the methods.