MarkovBin: An Algorithm to Cluster Metagenomic Reads Using a Mixture Modeling of Hierarchical Distributions

Tin Chi Nguyen
tin.nguyenchi@wayne.edu

Dongxiao Zhu
dzhu@wayne.edu

Department of Computer Science
Wayne State University
Detroit, MI 48202, USA

ABSTRACT

Metagenomics is the study of genomic content of microorganisms from environmental samples without isolation and cultivation. Recently developed next generation sequencing (NGS) technologies efficiently generate vast amounts of metagenomic DNA sequences. However, the ultra-high throughput and short read lengths make the separation of reads from different species more challenging. Among the existing computational tools for NGS data, there are supervised methods that use reference databases to classify reads and unsupervised methods that use oligonucleotide patterns to cluster reads. The former may leave a large fraction of reads unclassified due to the absence of closely related references. The latter often rely on long oligonucleotide frequencies and are sensitive to species abundance levels. In this work, we present MarkovBin, a new unsupervised method that can accurately cluster metagenomic reads across various species abundance ratios. We first model the nucleotide sequences as a fixed-order Markov chain. We then propose a hierarchical distribution to model the dependency between paired-end reads. Finally, we employ the mixture model framework to separate reads from different genomes in a metagenomic dataset. Using extensive simulation data, we demonstrate a high accuracy and precision by comparing to selected unsupervised read clustering tools. The software is freely available at `http://orleans.cs.wayne.edu/MarkovBin`.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Markov processes, Probabilistic algorithms; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms, Similarity measures*

General Terms

Algorithms

Keywords

Metagenomics, Mixture Model, Hierarchical Distribution

BCB, '13, September 22 - 25, 2013, Washington, DC, USA

1. INTRODUCTION

Since up to 99% of bacteria cannot be cultivated and thus are uncharacterized [1, 2] , metagenomics [3, 4] offers a huge advantage over the traditional methods of studying individual genomes, which rely on species isolation and cultivation. Successful projects have provided deeper insights into the microbial world by sequencing DNA material of microbial samples in their natural habitats, such as soil [5], acid drainage from an abandoned mine [6], sea water [7], and microbial communities of the human body [8, 9, 10]. Metagenomics opens up opportunities to uncover the diversity of the microbial worlds, their evolution, and their impacts on the environment and human health.

Sequencing technologies have rapidly evolved and have generated vast amounts of metagenomic data over the past few years. Compared to Sanger sequencing technology, the NGS technologies, such as Roche/454 [11] and Illumina/Solexa [12], have been greatly parallelized to produce millions to billions of reads with faster speed and lower cost, allowing for much greater sequencing depth. However, reads obtained from the NGS platforms are often very short. For example, Illumina generates reads with lengths 50-150 compared to 700-1000 bps in Sanger reads. Both the ultra high-throughput and the significantly shorter read lengths present serious challenges in metagenomic analysis.

One of the main goals of metagenomics is to identify the populations of microorganisms and the roles of individual microbes in their communities. There have been efforts to reconstruct the genomic sequences of all organisms in the sample using assembly tools. However, this task is complicated due to several factors. First, the number of species and their relative abundances are unknown and change from sample to sample. Second, sequencing data are usually fragmented and partial since environmental sequence sampling rarely produces all the sequences required for assembly. Finally, in addition to repetitive regions within individual genomes, genomes of closely related species may share homologous sequences, which lead into very complex repetitive patterns. Consequently, assembly of metagenomic sequences is generally very challenging [13, 14, 15]. For this reason, grouping reads into more homogenous sets of reads has become an important step in metagenomic analysis.

There are currently two classes of computational methods to separate metagenomic reads: supervised and unsupervised. In general, the supervised methods compare the sequenc-

ing reads against known references in the databases, such as known genomic sequences, genes, or proteins. Earlier methods often use taxonomic markers, e.g., 16S rRNA, rpoB, and recA, to assign reads into different groups [16, 17, 18, 19]. These methods are not applicable to the whole genome scale since only a single gene is used for read classification. More recent methods often align reads to known reference genomes or protein families and then assign reads to taxa [20, 21, 22, 23, 24]. Since most of the microbes found in the environment are uncharacterized, these methods may discard or misclassify a large fraction of reads due to the absence of closely related references.

In contrast to supervised methods, the unsupervised methods usually cluster reads from different species using DNA composition information. Pioneering unsupervised methods for Sanger reads of length 700-1000 bps are based on the fact that some composition properties, such as GC content and frequencies of short q-tuples are preserved across the same genome and varies greatly between different genomes [25, 26, 27]. Methods in this class include TETRA [26, 28], CompostBin [29], TACOA [30], LikelyBin [31], SCIMM [32], and MetaCluster 3.0 [33]. Since some DNA composition properties are preserved only in long fragments, the performance of these methods is influenced greatly by the length of the reads. The above tools were tested only on datasets with read length 700-1000 bps, for which they achieve reasonable performance.

AbundanceBin [34] is one of the first unsupervised methods to cluster NGS reads based on the frequencies of l-tuples (l is around 20). AbundanceBin assumes that the frequencies of these l-tuples, with an appropriate value of l, are linearly proportional to the genomic coverage. Consequently, the frequencies of l-tuples from multiple species are assumed to come from a mixture of Poisson distributions [35]. AbundanceBin estimates the Poisson parameters via the Expectation-Maximization (EM) algorithm and then clusters the metagenomic reads based on the frequencies of their l-tuples. The main limitation of AbundanceBin is that it cannot separate reads from species having similar abundance. Moreover, choosing the length of l-tuples is also a challenging task [35].

MetaCluster 4.0 [36] and MetaCluster 5.0 [37] are two recently developed unsupervised tools that can efficiently separate NGS reads from species having similar abundances. MetaCluster 4.0 consists of three steps. In step 1, it groups reads based on some longer w-tuples ($w \geq 35$) that are assumed to be unique for each genome. In step 2, it associates each read with a feature vector built from frequencies of short 4-tuples. It then applies the k-means algorithm to further divide the groups formed in step 1. In step 3, it merges the some of the groups based on the inner-cluster similarity. MetaCluster 4.0 was shown to perform well for species with similar abundances. However, the performance declines when the abundance levels of the species differ by large margins. In addition, the greedy k-means algorithm does not give a globally optimal solution, i.e., each execution of MetaCluster 4.0 might give a different clustering result.

Since MetaCluster 4.0 performs better when the abundance levels in a sample are more even, MetaCluster 5.0 [37] further refines the method by filtering reads from low abundance species. It first counts the frequencies of l-tuples and then filters reads with all l-tuples appearing at most T time (T is a pre-defined threshold). The goal of this filtering step is to separate reads into two groups: reads from low abundance species and reads from high abundance species. The two groups of reads are then clustered independently, resulting in a greatly improved performance. However, the initialization problem of the k-means algorithm and the task of choosing the length of l-tuples are still not thoroughly addressed. In addition, the accuracy of the method still relies on the evenness of the species abundances as we will demonstrate in the experimental results.

Here we present MarkovBin, a new clustering algorithm that can reliably cluster metagenomics short reads across various species abundance levels. The fundamental assumption of our method is that nucleotide sequences sampled from a genome follow a Markov model as has been used in many other methods [23, 27, 31, 32, 38, 39, 40]. For single-end reads, we use a Markov chain to model the nucleotide sequences within each cluster. For paired-end reads, we propose a hierarchical model to efficiently exploit the paired-end information, taking into account the fragment length distribution. Finally, we apply the EM algorithm to estimate the Markov parameters and relative species abundance. Using the estimated parameters, we assign each read to the cluster with the highest probability among all clusters. Via extensive simulation experiments, MarkovBin demonstrates high accuracy and precision regardless of species abundance levels.

2. METHODS

The GC content of bacterial genomes has been playing an important role for phylogenetic classification [41, 38]. The GC content can be conceptualized as zero-order Markov chains, in which the probability of observing a nucleotide does not depend on its preceding nucleotides. Higher order Markov chains have been successfully applied to capture genome signatures by modeling the distribution of nucleotide sequences [23, 27, 31, 32, 38, 39, 40]. In this section, we recapture the fixed-order Markov chains and then employ the mixture model to cluster metagenomic NGS reads by their species of origin.

2.1 Markov Models

2.1.1 Fixed-order Markov Chains

Given a set of DNA sequences sampled from a genome, we want to model the probability of occurrence of each sequence. We define a DNA sequence S of length w as a sequence of discrete variables $Y_1, Y_2 \ldots, Y_w$ on the alphabet $\mathcal{A} = \{A, T, C, G\}$. For any probabilistic model of sequences, the probability of S can be calculated as follows:

$$\begin{aligned} P(S) &= P(Y_1 Y_2 \ldots Y_w) \\ &= P(Y_1 Y_2 \ldots Y_{w-1}) P(Y_w | Y_1 Y_2 \ldots Y_{w-1}) \\ &= P(Y_1) P(Y_2 | Y_1) \ldots P(Y_w | Y_1 Y_2 \ldots Y_{w-1}) \end{aligned}$$

The key assumption of an m^{th} order Markov chain is that the distribution of any element Y_i in the sequence depends only on its m preceding elements (Figure 1). with this as-

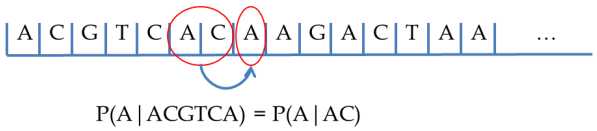

Figure 1: A second-order Markov chain. In this example, the probability of observing a nucleotide only depends only on its two preceding nucleotides.

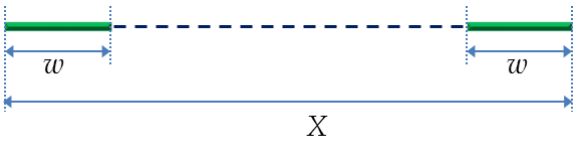

Figure 2: A paired-end read consisting of two short reads of length w.

sumption, the probability of S can be written as follows:

$$P(S) = P(Y_1 \dots Y_m) \prod_{i=m+1}^{w} P(Y_{i-m} \dots Y_{i-1}; Y_i) \quad (1)$$

Let us denote the parameter set of an m^{th} order Markov model as (ϑ, Φ), for which the transition probabilities are defined by $\Phi : \mathcal{A}^m \times \mathcal{A} \to [0,1]$ and the distribution of m-tuples is defined by $\vartheta : \mathcal{A}^m \to [0,1]$. The function Φ can be written using a transition probability matrix (TPM) of 4^m rows where each row has 4 possible transitions. Given the model (ϑ, Φ), we can compute the probability of $S = Y_1 Y_2 \dots Y_w$ as the product of the probability of the starting m-tuple and the transition probabilities as follows:

$$P(S|\vartheta, \Phi) = \vartheta(Y_1 \dots Y_m) \prod_{t=m+1}^{l} \Phi(Y_{t-m} \dots Y_{t-1}; Y_t) \quad (2)$$

2.1.2 Maximum Likelihood Estimators

Consider a set of n sequences $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$ independently generated by an m^{th} order Markov model (ϑ, Φ). The log likelihood of observing such data is given as:

$$\log L(\vartheta, \Phi|\mathcal{S}) = \sum_{i=1}^{n} \log P(S_i|\vartheta, \Phi) \quad (3)$$

where $P(S_i|\vartheta, \Phi)$ can be calculated as in (2). The maximum likelihood estimators (MLE) can be obtained as an appropriate root of the derivatives with respect to ϑ and Φ [42]. Solving the optimization problem yields the following maximum likelihood estimators:

$$\begin{aligned} \vartheta(c_{t-m} \dots c_{t-1}) &= \frac{N(c_{t-m} \dots c_{t-1})}{n(l-m+1)} \\ \Phi(c_{t-m} \dots c_{t-1}; c_t) &= \frac{N(c_{t-m} \dots c_{t-1} c_t)}{N(c_{t-m} \dots c_{t-1})} \end{aligned} \quad (4)$$

where $N(c_{t-m}, \dots c_{t-1})$ is the count of the $(m+1)$-tuple $c_{t-m} \dots c_{t-1}$ in $\mathcal{S}$, $c_{t-m}, \dots c_{t-1}$ is the count of the m-tuple $c_{t-m} \dots c_{t-1}$ in $\mathcal{S}$, and $n(l-m+1)$ is the total number of all m-tuple in $\mathcal{S}$. These MLE will be used in the M step of the EM algorithm in section 2.3.

2.2 Hierarchical Distributions

Sequencing technologies can generate a pair of short reads from both sides of a DNA fragment. Due to advanced size selection procedures, the distribution of the fragment lengths is approximately known. Using this information, we can extend the Markov models to calculate the distribution of paired-end data by modeling the dependency between a pair of short reads. Denoting X as the random variable corresponding to the fragment length of paired-end reads, w as the length of short reads (Figure 2), the probability of occurrence of a paired-end read $S = (S^1, S^2)$ can be calculated as follows:

$$\begin{aligned} P(S) &= \sum_{X \geq w} P(S^1, S^2|X) P(X) \\ &= \sum_{X > 2w} P(S^1, S^2|X) P(X) + \sum_{X=w}^{2w} P(S^1, S^2|X) P(X) \end{aligned}$$

When the two short reads are separated by a gap ($X > 2w$), the probability of occurrence of each nucleotide sequence can be treated as independent. Consequently, the first term of the above formula can be written as:

$$\begin{aligned} \sum_{X>2w} P(S^1, S^2|X) P(X) &= \sum_{X>2w} P(S^1|X) P(S^2|X) P(X) \\ &= \sum_{X>2w} P(S^1) P(S^2) P(X) \\ &= P(S^1) P(S^2) P(X \geq 2w) \end{aligned}$$

When the two short reads overlap ($X < 2w$) and form a contiguous sequence S_X ($|S_X| = X|$), we have $P(S^1, S^2|X) = P(S_X)$. Consequently, the probability of occurrence of the paired-end reads S can be calculated as follows:

$$P(S) = P(S^1) P(S^2) P(X \geq 2w) + \sum_{X=w}^{2w} P(S_X) P(X) \quad (5)$$

where $P(S^1), P(S^2), P(S_X)$ can be calculated according to the Markov model and $P(X)$ is given from the fragment length distribution.

The above formula requires end-users to provide the distribution of the fragment length, which may cause difficulties in practice. Here we develop some approximation formulas to calculate P(S) when the fragment length distribution is not available. When the average fragment length is much larger than twice of the read length, the second term of (5) is negligible and $P(X \geq 2w) \approx 1$. Consequently, the probability of occurrence of the paired-end read can be approximated as $P(S) = P(S^1)P(S^2)$. When the average fragment length is close to or smaller than twice of the read length, we can look for overlap between the two short reads. If they overlap, we can merge them into a single sequence S_X and then approximate $P(S)$ as $P(S) \approx P(S_X)$. Otherwise, we can approximate $P(S)$ as $P(S) \approx P(S^1)P(S^2)$.

2.3 Mixture Modeling for Cluster Analysis

In this section, we first apply the EM algorithm to cluster single-end reads using a mixture of Markov models [43, 44]. We then extend the framework to cluster paired-end reads using a mixture of hierarchical models. The workflow of the

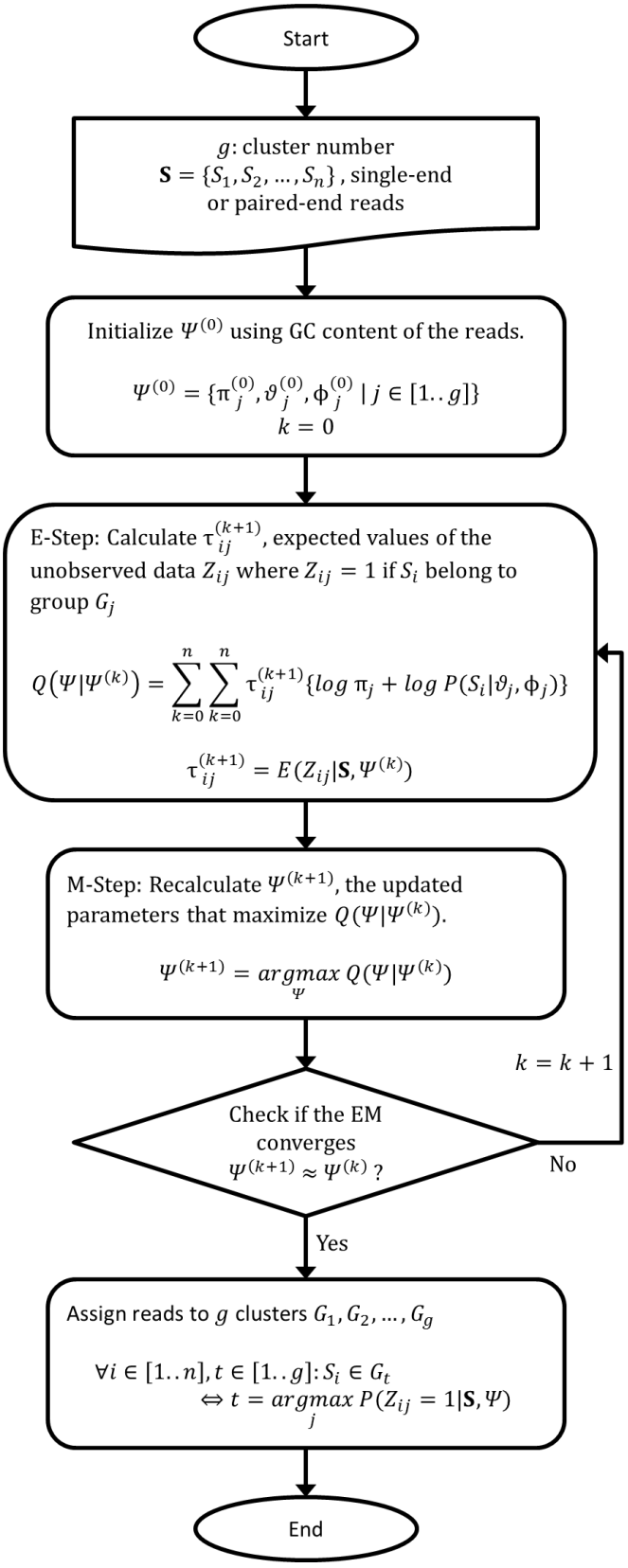

Figure 3: The workflow of MarkovBin.

MarkovBin is depicted in Figure 3, which will be described in details in the following sections.

2.3.1 Mixture of Markov Models

Consider a set of n sequences $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ that were sequenced from a sample of g species. In the mixture model, it is assumed that the data are from a mixture of g clusters with various proportions. Denoting (ϑ_j, Φ_j) and π_j as the Markov parameters and the proportion of the j^{th} cluster ($j \in [1..g]$), we have the following set of unknown parameters $\Psi = (\vartheta_1, \Phi_1, \pi_1, \ldots, \vartheta_g, \Phi_g, \pi_g)$. The goal of the algorithm is to estimate this set of parameters and then assign each read to the cluster with the highest probability.

The EM algorithm [45] is applied in the framework where each sequence S_i ($i \in [1..n]$) is considered to be observed and the indicator vector $\mathbf{z}_i$ denoting its source origin is considered to be missing. Each sequence S_i is associated with an unobserved indicator vector $\mathbf{z}_i = [z_{i1}, \ldots, z_{ig}]$, where $z_{ij} = 1$ if S_i is from the j^{th} cluster and $z_{ij} = 0$ otherwise ($i \in [1..n], j \in [1..g]$). The probability of a complete data point $(S_i, \mathbf{z}_i)$ is given by:

$$P(S_i, \mathbf{z}_i|\Psi) = \sum_{j=1}^{g} z_{ij}\pi_j P(S_i|\vartheta_j, \Phi_j) \tag{6}$$

where the probability $P(S_i|\vartheta_j, \Phi_j)$ can be calculated as described in (2). With the assumption that the sequences in $\mathcal{S}$ were independently generated from the mixture model, the complete data log likelihood $\log L_c(\Psi)$ is given by:

$$\begin{aligned}\log L_c(\Psi) &= \log(\prod_{i=1}^{n}\sum_{j=1}^{g} z_{ij}\pi_j P(S_i|\vartheta_j, \Phi_j)) \\ &= \sum_{i=1}^{n} \log(\sum_{j=1}^{g} z_{ij}\pi_j P(S_i|\vartheta_j, \Phi_j))\end{aligned}$$

In the above formula, $\mathbf{z}_i$ is an indicator vector, i.e., z_{ij} equals 1 only for one specific value of j and z_{ij} equals 0 for the rest. For that reason, we have the following equation:

$$\log(\sum_{j=1}^{g} z_{ij}\pi_j P(S_i|\vartheta_j, \Phi_j)) = \sum_{j=1}^{g} z_{ij} \log(\pi_j P(S_i|\vartheta_j, \Phi_j))$$

which leads to the following form of the complete data log likelihood:

$$\log L_c(\Psi) = \sum_{i=1}^{n}\sum_{j=1}^{g} z_{ij}\{\log \pi_j + \log P(S_i|\vartheta_j, \Phi_j)\} \tag{7}$$

The EM algorithm starts by an initialization of the unknown parameters and then iteratively recalculate them. Denoting the parameter set after k iterations as $\Psi^{(k)} = (\vartheta_1^{(k)}, \Phi_1^{(k)}, \pi_1^{(k)}, \ldots, \vartheta_g^{(k)}, \Phi_g^{(k)}, \pi_g^{(k)})$, the E step of the $(k+1)^{th}$ iteration calculates the conditional expectation of the complete data log likelihood given the observe data and the current values $\Psi^{(k)}$ of Ψ. Denoting Z_{ij} as the random variable corresponding to z_{ij} and $E(Z_{ij}|\mathcal{S}, \Psi^{(k)})$ as their expectation using the current value $\Psi^{(k)}$ of Ψ, the Q function is given by:

$$Q(\Psi|\Psi^{(k)}) = \sum_{i=1}^{n}\sum_{j=1}^{g} \tau_{ij}^{(k+1)}\{\log \pi_j + \log P(S_i|\vartheta_j, \Phi_j)\} \tag{8}$$

where $\tau_{ij}^{(k+1)}$ can be calculated as follows:

$$\begin{aligned}\tau_{ij}^{(k+1)} &= E(Z_{ij}|\mathcal{S},\Psi^{(k)}) \\ &= P(Z_{ij}=1|S_i,\Psi^{(k)}) \\ &= \frac{\pi_j^{(k)} P(S_i|\vartheta_j^{(k)},\Phi_j^{(k)})}{\sum_{t=1}^{g}\pi_t^{(k)} P(S_i|\vartheta_t^{(k)},\Phi_t^{(k)})}\end{aligned} \tag{9}$$

The M step of the $(k+1)^{th}$ iteration requires the maximization of the Q function with respect to Ψ over the parameter space to give the updated estimate $\Psi^{(k+1)}$. The updated estimate of the j^{th} proportion π_j is given by:

$$\pi_j^{(k+1)} = \sum_{i=1}^{n}\frac{\tau_{ij}^{(k+1)}}{n} \tag{10}$$

Using the maximum likelihood estimators in (4), the transition probabilities of the Markov models are given by the following equations:

$$\begin{aligned}\vartheta_j^{(k+1)}(c_{t-m},\dots c_{t-1}) &= \frac{N_j^{(k+1)}(c_{t-m},\dots c_{t-1})}{n(l-m+1)} \\ \Phi_j^{(k+1)}(c_{t-m},\dots c_{t-1};c_t) &= \frac{N_j^{(k+1)}(c_{t-m},\dots c_t)}{N_j^{(k+1)}(c_{t-m},\dots c_{t-1})} \\ N_j^{(k+1)}(c_{t-m},\dots c_t) &= \sum_i \tau_{ij}^{(k+1)} N_j(c_{t-m},\dots c_t) \\ N_j^{(k+1)}(c_{t-m},\dots c_{t-1}) &= \sum_i \tau_{ij}^{(k+1)} N_j(c_{t-m},\dots c_{t-1})\end{aligned} \tag{11}$$

The EM algorithm recalculate the parameters until they do not change over the new iteration. After the EM algorithm converges, we estimate the probability that a read S_i belongs to a cluster G_j as follows:

$$P(Z_{ij}=1|\mathcal{S},\Psi) = \frac{\pi_j P(S_i|\vartheta_j,\Phi_j)}{\sum_{t=1}^{g}\pi_t P(S_i|\vartheta_t,\Phi_t)} \tag{12}$$

Using (12), we assign each read to the cluster with the highest probability among all clusters. To increase the confidence level, we can set a cutoff to remove ambiguity. If the maximum value is distinctively higher than the rest, we can confidently assign the read into the corresponding cluster, otherwise we can just discard the read. By default, the filtering cutoff is not set, i.e., no read is discarded.

2.3.2 Mixture of Hierarchical Models

Consider a sample $\mathcal{S} = \{(S_1^1,S_1^2),(S_2^1,S_2^2),\dots,(S_n^1,S_n^2)\}$ of n paired-end reads originated from g clusters of different hierarchical models. We can follow the same workflow as explained in section 2.3.1 using the new forms of the complete log likelihood and the Q function:

$$\log L_c(\Psi) = \sum_{i=1}^{n}\sum_{j=1}^{g} z_{ij}\{\log\pi_j + \log P(S_i^1,S_i^2|\vartheta_j,\Phi_j)\}$$

$$Q(\Psi|\Psi^{(k)}) = \sum_{i=1}^{n}\sum_{j=1}^{g} \tau_{ij}^{(k+1)}\{\log\pi_j + \log P(S_i^1,S_i^2|\vartheta_j,\Phi_j)\}$$

$$\tau_{ij}^{(k+1)} = \frac{\pi_j^{(k)} P(S_i^1,S_i^2|\vartheta_j^{(k)},\Phi_j^{(k)})}{\sum_{t=1}^{g}\pi_t^{(k)} P(S_i^1,S_i^2|\vartheta_t^{(k)},\Phi_t^{(k)})}$$

where $P(S_i^1,S_i^2|\vartheta_j^{(k)})$ can be calculated using (5).

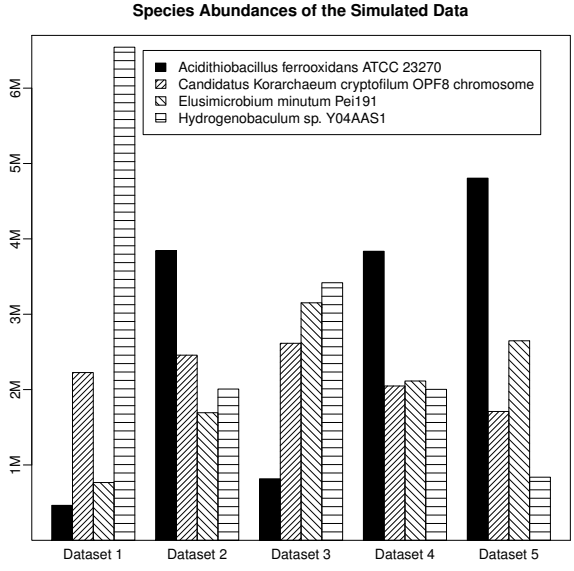

Figure 4: Species abundances of the simulated data. The horizontal axis displays names of the 5 datasets whereas the vertical axis displays the number of reads from each species. Each dataset consists of 10 million paired-end reads with length 100.

2.3.3 Initial Values for the Mixture Model

The choice of initial values is essential for our algorithm because of the tendency for there to exist many local maxima of the likelihood function. One typical solution is to run the algorithm multiple times with random initial values and then choose the best clustering result according to the complete log likelihood. However, due to the large number of parameters, this approach has raised concerns about the repeatability and the computational demand.

To avoid exploring the vast parameter space, we have developed a new procedure to initialize the Markov models using GC content. We first count the number of G and C nucleotides in each read. We then sort the reads according to their G and C counts before evenly dividing them into g clusters. Using the reads falling into each cluster, we calculate the initial values for the mixture model.

3. SIMULATION EXPERIMENTS

We used MetaSim [46], an open source software package, to generate 5 datasets with various abundances from 4 different species: *"Acidithiobacillus ferrooxidans ATCC 23270"*, *"Candidatus Korarchaeum cryptofilum OPF8 chromosome"*,

Table 1: Running time of MarkovBin, AbundanceBin, and MetaCluster (in minutes).

Datasets	MarkovBin	AbundanceBin	MetaCluster
1	625	376	112
2	195	263	55
3	173	319	69
4	193	226	56
5	237	262	56

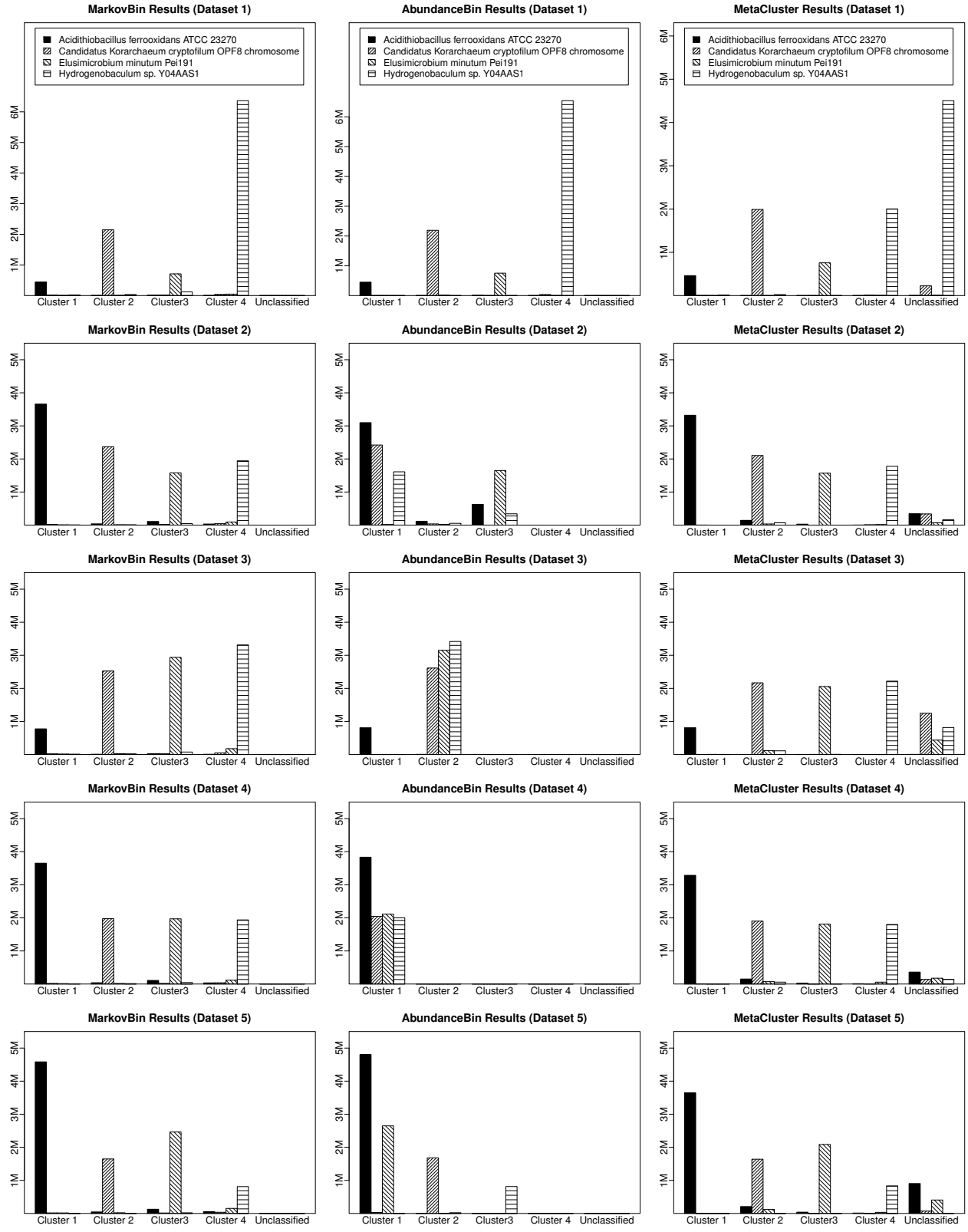

Figure 5: Clustering results of MarkovBin (*left* panels), AbundanceBin (*middle* panels), and MetaCluster (*right* panels) using simulated data. The horizontal axes display the clusters whereas the vertical axes display the read numbers from different species classified into each cluster. The last cluster in each panel displays the number of unclassified reads.

Table 2: The accuracy, precision, and adjusted Rand index of MarkovBin, AbundanceBin, and MetaCluster.

Datasets	MarkovBin			AbundanceBin			MetaCluster		
	Accuracy	Precision	ARI	Accuracy	Precision	ARI	Accuracy	Precision	ARI
1	0.94	0.97	0.91	**0.99**	**0.99**	**0.98**	0.60	0.96	0.56
2	**0.91**	0.97	**0.89**	0.78	0.49	0.20	0.77	0.97	0.78
3	0.91	**0.96**	**0.88**	**1**	0.21	0.13	0.58	0.93	0.55
4	0.91	0.97	**0.89**	**1**	0	0	0.77	0.97	0.78
5	0.90	0.97	**0.89**	**0.99**	0.61	0.51	0.65	0.97	0.66

"Elusimicrobium minutum Pei191", and *"Hydrogenobaculum sp. Y04AAS1"*. The GC content of the species are 58%, 55%, 51%, and 35%, respectively. Each dataset consists of 10 million paired-end reads with length 100. The species abundances are displayed in Figure 4.

We compare the performance of MarkovBin with AbundanceBin (version 1.01, February 2013) and MetaCluster (version 5.0, May 2012), which are 2 of the latest unsupervised binning tools for NGS short reads. For all the 3 methods, we set the the number of species and the read length set to 4 and 100, respectively. For MarkovBin, we set the Markov order to 1. Table 1 shows the time complexity of the competing tools on the same server (4 x Twelve-Core AMD Opteron 2.6GHz, 256GB RAM). For all the 5 data sets, MetaCluster runs much faster than AbundanceBin and MarkovBin. Interestingly, the running time of all the three methods increases significantly for the first dataset, which indicates a slow conversion when species abundances differ by large margins. Overall, all the three clustering methods are reasonably fast and can cluster 10 million paired-end reads in several hours.

We proceed to compare the performance of the clustering methods. Each tool outputs 4 clusters of reads and a set of unclassified reads (discarded reads). We treat the set of unclassified reads as the 5^{th} cluster. For MetaCluster 5.0, we did not proceed to the second round (MetaCluster5_2) to further cluster the unclassified reads because the number of clusters already exceeds the number of species. In Figure 5, the left panels show the clustering results of MarkovBin whereas the middle panels and right panels show the clustering results of AbundanceBin and MetaCluster. In each panel, the horizontal axis displays the cluster names whereas the vertical axis displays the read number of each species falling into each cluster. The last clusters in the panels consist of the reads that were discarded. In general, MarkovBin and MetaCluster perform consistently well regardless of species abundance with the difference is that MetaCluster discards a large fraction of reads. On the other hand, AbundanceBin gives a good result only when the species have very different relative abundances.

To give more comprehensive comparisons, we also calculate the accuracy and precision of the competing methods. We consider a pair of reads to be positive if they belong to the same species. Likewise, we consider a pair of reads to be negative if they belong to different species. Let us denote N_P as the total number of the positive pairs, N_N as the total number or the negative pairs, N_{TP} (true positive) as the number of positive pairs that were assigned to the same cluster, N_{TN} (true negative) as the number of negative pairs that were assigned to different clusters. We define the accuracy as $\frac{N_{TP}}{N_P}$ and the precision as $\frac{N_{TN}}{N_N}$. The accuracy and precision of each method are shown in Table 2. To access the overall performance, we also report the adjusted Rand index (ARI) [47], which is the corrected-for-chance version of the Rand index [48].

In general, AbundanceBin achieves high accuracy as it usually groups reads of the same species together but cannot separate reads from species having similar genomic coverage, resulting in a fluctuating precision. MetaCluster offers high precision across different abundance ratios since it relies on the composition properties, i.e., long w-tuples ($w \geq 35$) and short 4-tuples, to separate the reads. However, its performance relies on the evenness of the species abundance in the sample. To make the species abundance more even, MetaCluster discarded a lot of reads in the filtering step.

Compared to AbundanceBin and MetaCluster, MarkovBin has a more stable performance since it consistently yields high accuracy (> 0.9) and precision (> 0.96) for all of the 5 simulated datasets. There are three possible reasons for this. First, the GC content generally captures the genome signatures and thus provides good initial values for the EM algorithm. Second, our model is not affected by species abundance since it relies only on genomic composition. Finally, the hierarchical model efficiently exploits the paired-end information. It connects the two nucleotides sequences of any paired-end read together, resulting in a more stable feature extraction.

4. CONCLUSIONS

In this paper, we have presented MarkovBin, a new algorithm to cluster metagenomic NGS reads. The mixture model based algorithm is expected to give better clustering performance over the k-means algorithm whereas the hierarchical model allows for formulating the paired-end information. We also provided a robust initialization using GC content. We evaluated and compared the performance of MarkovBin with selected tools using multiple simulated datasets. Overall, MarkovBin consistently achieves high precision and accuracy across various species abundance ratios.

For future work, we hope to develop a framework to combine different complementary features for a better clustering. Existing approaches, including AbundanceBin and MetaCluster, have provided very different features, which may play critical roles for separating reads under some specific conditions. We also hope to extend our method to automatically determine the number of species via some information

criteria, such as Akaike information criterion (AIC) [49] or Bayesian information criterion (BIC) [50]. Finally, we hope to assess the potential of our method by applying MarkovBin on real metagenomic samples with high species diversity to gain insights into complex microbial communities.

5. REFERENCES

[1] M. S. Rappé and S. J. Giovannoni. The uncultured microbial majority. *Annual Reviews in Microbiology*, 57(1):369–394, 2003.

[2] J. A. Eisen. Environmental shotgun sequencing: its potential and challenges for studying the hidden world of microbes. *PLoS biology*, 5(3):e82, 2007.

[3] J. Handelsman, M. R. Rondon, S. F. Brady, J. Clardy, and R. M. Goodman. Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chemistry & Biology*, 5(10):R245–R249, 1998.

[4] K. Chen and L. Pachter. Bioinformatics for whole-genome shotgun sequencing of microbial communities. *PLoS computational biology*, 1(2):e24, 2005.

[5] S. Leininger, T. Urich, M. Schloter, L. Schwark, J. Qi, G. Nicol, J. Prosser, S. Schuster, and C. Schleper. Archaea predominate among ammonia-oxidizing prokaryotes in soils. *Nature*, 442(7104):806–809, 2006.

[6] G. W. Tyson, J. Chapman, P. Hugenholtz, E. E. Allen, R. J. Ram, P. M. Richardson, V. V. Solovyev, E. M. Rubin, D. S. Rokhsar, and J. F. Banfield. Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature*, 428(6978):37–43, 2004.

[7] S. Yooseph, G. Sutton, D. B. Rusch, A. L. Halpern, S. J. Williamson, K. Remington, J. A. Eisen, K. B. Heidelberg, G. Manning, W. Li, et al. The sorcerer ii global ocean sampling expedition: expanding the universe of protein families. *PLoS biology*, 5(3):e16, 2007.

[8] E. K. Costello, C. L. Lauber, M. Hamady, N. Fierer, J. I. Gordon, and R. Knight. Bacterial community variation in human body habitats across space and time. *Science*, 326(5960):1694–1697, 2009.

[9] E. A. Grice, H. H. Kong, S. Conlan, C. B. Deming, J. Davis, A. C. Young, G. G. Bouffard, R. W. Blakesley, P. R. Murray, E. D. Green, et al. Topographical and temporal diversity of the human skin microbiome. *science*, 324(5931):1190–1192, 2009.

[10] J. Qin, R. Li, J. Raes, M. Arumugam, K. S. Burgdorf, C. Manichanh, T. Nielsen, N. Pons, F. Levenez, T. Yamada, et al. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65, 2010.

[11] M. Margulies, M. Egholm, W. E. Altman, S. Attiya, J. S. Bader, L. A. Bemben, J. Berka, M. S. Braverman, Y.-J. Chen, Z. Chen, et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, 2005.

[12] D. R. Bentley. Whole-genome re-sequencing. *Current opinion in genetics & development*, 16(6):545–552, 2006.

[13] M. Pop. Genome assembly reborn: recent computational challenges. *Briefings in bioinformatics*, 10(4):354–366, 2009.

[14] A. Charuvaka and H. Rangwala. Evaluation of short read metagenomic assembly. *BMC genomics*, 12(Suppl 2):S8, 2011.

[15] H. Teeling and F. O. Glöckner. Current opportunities and challenges in microbial metagenome analysisÛa bioinformatic perspective. *Briefings in bioinformatics*, 13(6):728–742, 2012.

[16] D. J. Lane, B. Pace, G. J. Olsen, D. A. Stahl, M. L. Sogin, and N. R. Pace. Rapid determination of 16s ribosomal rna sequences for phylogenetic analyses. *Proceedings of the National Academy of Sciences*, 82(20):6955–6959, 1985.

[17] J. Cole, B. Chai, R. Farris, Q. Wang, S. Kulam, D. McGarrell, G. Garrity, and J. Tiedje. The ribosomal database project (rdp-ii): sequences and tools for high-throughput rrna analysis. *Nucleic Acids Research*, 33(suppl 1):D294–D296, 2005.

[18] S. Chakravorty, D. Helb, M. Burday, N. Connell, and D. Alland. A detailed analysis of 16s ribosomal rna gene segments for the diagnosis of pathogenic bacteria. *Journal of microbiological methods*, 69(2):330–339, 2007.

[19] R. J. Case, Y. Boucher, I. Dahllöf, C. Holmström, W. F. Doolittle, and S. Kjelleberg. Use of 16s rrna and rpob genes as molecular markers for microbial ecology studies. *Applied and environmental microbiology*, 73(1):278–288, 2007.

[20] A. C. McHardy, H. G. Martin, A. Tsirigos, P. Hugenholtz, and I. Rigoutsos. Accurate phylogenetic classification of variable-length dna fragments. *Nature methods*, 4(1):63–72, 2006.

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

[22] M. Wu and J. A. Eisen. A simple, fast, and accurate method of phylogenomic inference. *Genome Biology*, 9(10):R151, 2008.

[23] A. Brady and S. L. Salzberg. Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nature methods*, 6(9):673–676, 2009.

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

[25] T. Abe, S. Kanaya, M. Kinouchi, Y. Ichiba, T. Kozuki, and T. Ikemura. Informatics for unveiling hidden genome signatures. *Genome Res.*, 13(4):693–702, 2003.

[26] H. Teeling, A. Meyerdierks, M. Bauer, R. Amann, and F. O. Glockner. Application of tetranucleotide frequencies for the assignment of genomic fragments. *Environmental Microbiology*, 6(9):938–947, 2004.

[27] J. Bohlin, E. Skjerve, and D. W. Ussery. Investigations of oligonucleotide usage variance within and between prokaryotes. *PLoS computational biology*, 4(4):e1000057, 2008.

[28] H. Teeling, J. Waldmann, T. Lombardot, M. Bauer, and F. O. Glöckner. TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. *BMC Bioinformatics*, 5(1):163, 2004.

[29] S. Chatterji, I. Yamazaki, Z. Bai, and J. A. Eisen. CompostBin: A DNA composition-based algorithm for binning environmental shotgun reads. In *Research in Computational Molecular Biology*, pages 17–28. Springer, 2008.

[30] N. N. Diaz, L. Krause, A. Goesmann, K. Niehaus, and T. W. Nattkemper. TACOA–Taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach. *BMC bioinformatics*, 10(1):56, 2009.

[31] A. Kislyuk, S. Bhatnagar, J. Dushoff, and J. Weitz. Unsupervised statistical clustering of environmental shotgun sequences. *BMC bioinformatics*, 10(1):316, 2009.

[32] D. Kelley and S. Salzberg. Clustering metagenomic sequences with interpolated Markov models. *BMC Bioinformatics*, 11(1):544, 2010.

[33] H. C. Leung, S. Yiu, B. Yang, Y. Peng, Y. Wang, Z. Liu, J. Chen, J. Qin, R. Li, and F. Y. Chin. A robust and accurate binning algorithm for metagenomic sequences with arbitrary species abundance ratio. *Bioinformatics*, 27(11):1489–1495, 2011.

[34] Y.-W. Wu and Y. Ye. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. *Journal of Computational Biology*, 18(3):523–534, 2011.

[35] X. Li and M. S. Waterman. Estimating the Repeat Structure and Length of DNA Sequences Using l-Tuples. *Genome research*, 13(8):1916–1922, 2003.

[36] Y. Wang, H. C. Leung, S. Yiu, and F. Y. Chin. Metacluster 4.0: a novel binning algorithm for ngs reads and huge number of species. *Journal of Computational Biology*, 19(2):241–249, 2012.

[37] Y. Wang, H. C. Leung, S. Yiu, and F. Y. Chin. MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. *Bioinformatics*, 28(18):i356–i362, 2012.

[38] J. C. Wooley, A. Godzik, and I. Friedberg. A primer on metagenomics. *PLoS computational biology*, 6(2):e1000667, 2010.

[39] S. Schbath, B. Prum, and E. DE TURCKHEIM. Exceptional motifs in different Markov chain models for a statistical analysis of DNA sequences. *Journal of Computational Biology*, 2(3):417–437, 1995.

[40] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[41] L. G. Wayne. International committee on systematic bacteriology: announcement of the report of the ad hoc committee on reconciliation of approaches to bacterial systematics. *Systematic and Applied Microbiology*, 10(2):99–100, 1988.

[42] S. Schbath, B. Prum, and E. de Turckheim. Exceptional motifs in different markov chain models for a statistical analysis of dna sequences. *Journal of Computational Biology*, 2(3):417–437, 1995.

[43] G. J. McLachlan and S. U. Chang. Mixture modelling for cluster analysis. *Statistical Methods in Medical Research*, 13(5):347–361, 2004.

[44] J.-J. Daudin, S. Li-Thiao-Te, and E. Lebarbier. Statistical challenges from the analysis of NGS-Metagenomics experiment., 2010.

[45] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*, volume 382. John Wiley & Sons, 2007.

[46] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson. MetaSim-A Sequencing Simulator for Genomics and Metagenomics. *PLoS ONE*, 3(10):e3373, 2008.

[47] L. Hubert and P. Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[48] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[49] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[50] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.