Poisson-Markov Mixture Model and Parallel Algorithm for Binning Massive and Heterogenous DNA Sequencing Reads

Lu Wang, Dongxiao Zhu(✉), Yan Li, and Ming Dong

Department of Computer Science, Wayne State University,
Detroit, MI 48202, USA
{lu.wang3,dzhu,rock_liyan,mdong}@wayne.edu

Abstract. A major computational challenge in analyzing metagenomics sequencing reads is to identify unknown sources of massive and heterogeneous short DNA reads. A promising approach is to efficiently and sufficiently extract and exploit sequence features, i.e., k-mers, to bin the reads according to their sources. Shorter k-mers may capture base composition information while longer k-mers may represent reads abundance information. We present a novel Poisson-Markov mixture Model (PMM) to systematically integrate the information in both long and short k-mers and develop a parallel algorithm for improving both reads binning performance and running time. We compare the performance and running time of our PMM approach with selected competing approaches using simulated data sets, and we also demonstrate the utility of our PMM approach using a time course metagenomics data set. The probabilistic modeling framework is sufficiently flexible and general to solve a wide range of supervised and unsupervised learning problems in metagenomics.

Keywords: Probabilistic clustering · Expectation-Maximization algorithm · Metagenomics · Next-generation sequencing (NGS) · Parallel algorithm

1 Introduction

Metagenomics sequencing reads are typically sequenced from a large number of heterogeneous sources with diverse abundances. There are two related yet distinct computational problems. The first is unsupervised binning of the reads to identify unknown sources. Reads from the same sources are more similar compared to the rest and the sources can later be labeled as Operational Taxonomic Units (OTU's). The other is supervised classification of the reads to assign each read to a labeled known source, such as a taxonomic or a patient treatment/risk group. Here we will focus on the more challenging reads binning problem.

Reads binning has posed the following unprecedented algorithmic and computational challenges, ranked by decreasing priority, to bioinformatics research

A. Bourgeois et al. (Eds.): ISBRA 2016, LNBI 9683, pp. 15–26, 2016.
DOI: 10.1007/978-3-319-38782-6_2

community: (1) How to sufficiently and robustly extract discriminating features from the reads? This is essentially a k-mers (sequencing feature) counting and selection problem; (2) How to account for the differential abundances across bins? Some sources may generate more reads whereas others may generate less; (3) How to filter out the inseparable reads? Some reads contain useful feature information, but others don't. The latter can come from the common sequences shared among the sources and was referred to as inseparable reads, and (4) How to efficiently process ultra-high throughput (hundreds of millions), very short ($\approx$ 100 bp) reads?

A key to overcome the first challenge is to sufficiently and robustly extract sequence features, i.e., k-mers (substring of length k), from NGS reads since it is the only information available from DNA sequencing data. Earlier approaches usually align the entire reads to non-redundant coding sequences (nr) and/or functional groups based on sequence similarity, usually via a BLASTX search. In metagenomics, familiar examples include CARMA [4], MEGAN [6] and Phymm [1]. CARMA attempts to assign short reads to known Pfam domains (structural components conserved across multiple proteins) and protein families [4]. MEGAN classifies reads to the Lowest Common Ancestor (LCA) based on multiple BLASTX score hits [6]. These dynamic programming approaches use information in the long k-mers to construct optimal read sequence alignment result.

Other approaches used information in the shorter k-mers. Phymm used interpolated Markov models (IMMs) [18] to characterize variable-length short k-mers that are typical of a phylogenetic grouping. Short k-mers, such as oligonucleotide [14], dinucleotide [7] and tetranucleotide counts [16,19], were used as the discriminative features to capture the information on base composition heterogeneity, perhaps in deference to the long sequencing contigs generated from the earlier sequencing technology. In particular, our recent work [16] used short k-mers in a mixture of Markov chains to calculate the probability of each read assigned to each bin. Presumably, reads binning approaches using both short k-mers and long k-mers as features are more desirable.

An effective approach to overcome the second challenge is to explicitly capture abundance information. For example, AbundanceBin extracted and used feature information from long k-mers of the reads, which directly yield read abundance information [21], to fit a mixture of Poisson models. Each component models the abundance of an individual bin. Similarly, an effective approach to overcome the third challenge is to develop non-mutually exclusive probabilistic clustering methods, where each read can simultaneously fall into different clusters with different posterior probabilities. A read with similar posterior probabilities across all the bins can be considered as non-informative, thus inseparable reads.

Due to the increasing degree of problem complexity, recent works focused more on developing analytic workflows, which exploit the information in short and/or long k-mers and solve the problem in a heuristic manner, e.g., [9,20]. However, the short and long k-mer reads features were not used systematically, i.e., the performance can be compromised by the choices of user-defined cut-off's and the heuristic k-means type algorithms. Thus, it is subject to high variance. Moreover, the deterministic reads partitioning significantly undermines

performance, especially for the inseparable reads that are sequenced from the common and/or low-complexity regions of the meta-genomes.

Therefore, it is desirable to develop a systematic approach to robustly and sufficiently integrate reads base composition information and reads abundance information into a single probability model to maximize the binning performance. By assuming these two pieces of information are captured by short k-mers and long k-mers, respectively, we propose a novel Poisson-Markov Model (PMM) approach to integrate reads feature information for binning and classifying short reads. Specifically, we extract reads feature information in both short k-mers and long k-mers to combat the outstanding issues of read heterogeneity and abundance variation in short DNA sequencing reads. We use probability models to accommodate the uncertainties and errors in reads assignment, and we develop a joint mixture model to systematically integrate sequencing feature information. Additionally, our joint mixture model overcomes the third challenge by adopting a soft reads binning, which enables a better performance by filtering out inseparable reads, e.g., those from orthologs or introns across genomes.

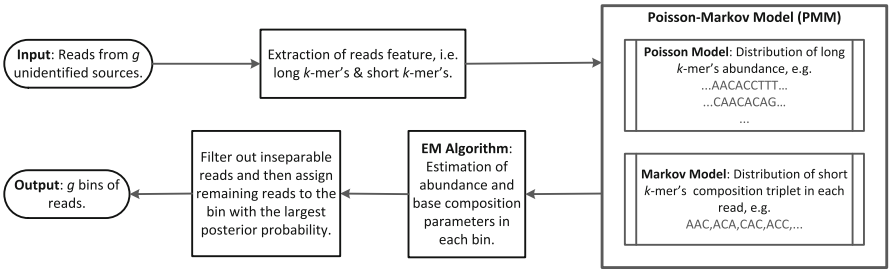


Fig. 1. A conceptual overview of the Poisson-Markov modeling approach for binning of DNA sequencing reads.

We claim that it is one-of-the-kind probabilistic modeling approaches to integrate feature information for binning and classifying short DNA sequencing reads. PMM has been applied in a number of different areas to solve a wide range of problems arising in biomedical science [12], animal science [11], agriculture science [8] and actuarial science [3]. By exploiting efficient data structures for counting k-mers and parallelizing likelihood calculations to multiple threads, we overcome the fourth challenge and make our binning approach more scalable to ever-increasing data volume. Figure 1 presents the main idea of this work.

2 Method

2.1 Poisson-Markov Model (PMM)

We assume a set of n DNA sequencing reads are sampled from g bins with N sequencing reads from each bins. A DNA sequence read is defined as S with

discrete variables y_i from $\{A, T, C, G\}$. We also assume reads abundance in j^{th} bin follows a Poisson distribution with parameter λ_j and the reads base composition in the bin is calculated by a Markov model with parameter τ. Please refer to Table 1 for the list of mathematical symbols used in this paper. A joint probability model $f(y_i)$ is shown as:

$$f(y_i) = P(k_j|\lambda_j)P(y_i \mid \tau), \tag{1}$$

where i represents read index and j represents bin index. Assuming there are k_j sequences in j^{th} bin, so the abundance of j^{th} bin can be shown in Poisson as:

$$P(k_j|\lambda_j) = \frac{\lambda_j^{k_j} e^{-\lambda_j}}{k_j!}. \tag{2}$$

In order to develop a probability model for binning and classification of DNA sequencing reads, we need to introduce another variable Z_{ij}, where $Z_{ij} = 1$ means the sequencing read S_i belongs to j^{th} bin, otherwise not. Z_{ij} is given (as the label)

Table 1. A list of mathematical symbols

Notations	Comments
n	number of DNA sequence reads
N	number of DNA sequence reads in each bin
S	a DNA sequencing read
i	index of the reads $\in [1, ..., n]$
S_i	i^{th} sequencing read in given dataset
y_i	discrete variables A, T, C, G
g	number of bins
j	index of the bins $\in [1, ..., g]$
τ	latent variable of Poisson-Markov Model
k_j	number of reads in j^{th} bin
λ_j	parameter of Poisson Model in j^{th} bin
Z_{ij}	indicator whether read S_i belongs to j^{th} bin
ϕ_j	4 by 4^m Transition Probability Matrix
m	tuple/order of TPM
π_j	proportion of j^{th} tbin
c	G/C count
Θ	parameter of Poisson-Markov Model
τ_{ij}	posterior binning probability
l	index of the iterations
P_x	x^{th} partition in parallel computing of E-step
x	number of partitions in parallel computing of E-step

in supervised classification problems whereas it is a latent variable in unsupervised binning problem. Therefore, we focus on the more challenging read binning problem and applications to solve reads classification problem as follows.

In a Markov model, Transition Probability Matrix (TPM) is represented with parameter ϕ, and π_j is the initial proportion of j^{th} bin. When Z_{ij}=1, the probability of a sequencing read S_i belongs to j^{th} bin is:

$$P(y_i \mid \tau) = P(Z_{ij} = 1 \mid S) = P(S_i \mid \phi_j), \tag{3}$$

and

$$P(S_i \mid \phi) = \sum_{j=1}^{g} \pi_j P(S_i \mid \phi_j). \tag{4}$$

$P(S_i \mid \phi_j)$ is the probability of observing read S_i which can be calculated using counts of the k-mers. ϕ_j is the TPM of 4 by 4^m calculated as:

$$\phi_j(c_{t-m} \cdots c_{t-1} c_t) = \frac{N(c_{t-m} \cdots c_{t-1} c_t)}{N(c_{t-m} \cdots c_{t-1})}, \tag{5}$$

where m is the tuple of TPM. $N(c_{t-m}, \ldots c_{t-1} c_t)$ is the count of the $(m+1)$-tuple, i.e., $c_{t-m} \ldots c_{t-1} c_t$, in $\mathcal{S}$ and $c_{t-m}, \ldots c_{t-1}$ is the count of the m-tuple $N(c_{t-m} \ldots c_{t-1})$ in $\mathcal{S}$. For example, in a second-order Markov model, ϕ_j is the TPM using a 4 by 16 probability matrix, where m and t equal to 2 and 3 respectively, which can be calculated by counting the corresponding 3-mers. Please see [16] for further details in calculating $P(S_i \mid \phi_j)$.

The complete data log-likelihood of Poisson-Markov Model $L_c(\Theta)$ can be written as:

$$\begin{aligned} \log L_c(\Theta) &= \log \left(\prod_{i=1}^{n} \sum_{j=1}^{g} Z_{ij} \frac{\lambda_j^{k_j} e^{-\lambda_j}}{k_j!} \pi_j P(S_i \mid \phi_j) \right) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{g} Z_{ij} \{ \log \lambda_j^{k_j} - \lambda_j - \log k_j! + \log \pi_j \\ &\quad + \log P(S_i \mid \phi_j) \}. \end{aligned} \tag{6}$$

The expected value of Z_{ij} is τ_{ij}, where Z_{ij} is a latent variable indicating whether the read i belongs to j^{th} bin:

$$\begin{aligned} \tau_{ij} &= E[Z_{ij} = 1 \mid \pi_j S, \phi] = P(Z_{ij} = 1 \mid \pi_j S_i, \phi_j) \\ &= \frac{P(N = k_j) \pi_j P(S_i \mid \phi_j)}{\sum_{j=1}^{g} P(N = k_j) \pi_j P(S_i \mid \phi_j)}. \end{aligned} \tag{7}$$

2.2 An Expectation-Maximization Algorithm

Here we develop an Expectation-Maximization (EM) algorithm to maximize the complete data log-likelihood function $\log L_c(\Theta)$. In the E-step, we calculate the

expected values of the log-likelihood function $\log L_c(\Theta)$, i.e., $Q(\Theta \mid \Theta^{(l)})$, under the current estimate of the parameters $\Theta^{(l)}$ in l^{th} iteration, where $\Theta = (\lambda_j, \tau)$, the set of parameters in Poisson and Markov models.

$$Q(\Theta \mid \Theta^{(l)}) = \sum_{i=1}^{n}\sum_{j=1}^{g} \tau_{ij}^{(l+1)} \{\log^{(l)} \lambda_j^{k_j} - \lambda_j^{(l)} - \log^{(l)} k_j! + \log^{(l)} \pi_j + \log P(S_i \mid \phi_j)\}.$$

In the M-step, we find the parameter values that maximize the $Q(\Theta \mid \Theta^{(l)})$. Specifically, τ_{ij} after $l+1$ iterations is calculated as:

$$\begin{aligned} \tau_{ij}^{(l+1)} &= E[Z_{ij} = 1 \mid \pi_j^{(l)} S, \phi^{(l)}] = P(Z_{ij} = 1 \mid \pi_j^{(l)} S_i, \phi_j^{(l)}) \\ &= \frac{P(N = k_j)^{(l)} \pi_j^{(l)} P(S_i \mid \phi_j^{(l)})}{\sum_{j=1}^{g} P(N = k_j)^{(l)} \pi_j^{(l)} P(S_i \mid \phi_j^{(l)})}. \end{aligned} \tag{8}$$

π_j is the proportion of j^{th} bin, so that π_j is updated by summarizing the expected counts of reads as:

$$\pi_j^{(l+1)} = \sum_{i=1}^{n} \frac{\tau_{ij}^{(l+1)}}{n}. \tag{9}$$

$\phi_j^{(l+1)}$ is the second-order TPM which can be updated as in [16]:

$$\begin{aligned} \phi_j^{(l+1)}(c_{t-m} \ldots c_{t-1} c_t) &= \frac{N_j^{(l+1)}(c_{t-m} \ldots c_{t-1} c_t)}{N_j^{(l+1)}(c_{t-m} \ldots c_{t-1})}, \\ N_j^{(l+1)}(c_{t-m} \ldots c_{t-1} c_t) &= \sum_{i=1}^{n} \tau_{ij}^{(l+1)} N_j(c_{t-m} \ldots c_{t-1} c_t), \\ N_j^{(l+1)}(c_{t-m} \ldots c_{t-1}) &= \sum_{i=1}^{n} \tau_{ij}^{(l+1)} N_j(c_{t-m} \ldots c_{t-1}). \end{aligned}$$

λ_j is estimated by calculating the first derivative of $Q(\Theta \mid \Theta^{(l)})$ as:

$$\frac{dQ(\Theta \mid \Theta^{(l)})}{d\lambda_j} = 0. \tag{10}$$

Thus we have:

$$\lambda_j^{(l+1)} = k_j^{(l+1)}. \tag{11}$$

The E and M steps alternates until convergence.

2.3 A Parallel Implementation of the PMM Algorithm

The E-step calculates the expected values of complete data log-likelihood which can be calculated using multiple threads in parallel where each thread calculates

Algorithm 1. The Parallelized PMM Algorithm

Input: n DNA sequencing reads $S = S_1, ..., S_i, ..., S_n$, Number of clusters g.

for $j = 1$ *to* g **do**
  Initialize $\Theta^{(0)}$:
    $\pi_j = \frac{1}{g}$, $k_j = \frac{n}{g}$, $\phi_j(c_{t-m} \cdots c_{t-1}c_t) = \frac{N(c_{t-m} \cdots c_{t-1}c_t)}{N(c_{t-m} \cdots c_{t-1})}$ and $\lambda_j = \frac{n}{g}$;
end
repeat
  E-step: Compute the responsibilities at l^{th} iteration
    Distribute the log-likelihood table ($n \times g$) into x partitions for parallel computation;
    $\hat{\tau}_{ij} = E[Z_{ij} = 1 \mid \pi_j, S, \phi] = p(Z_{ij} = 1 \mid \pi_j, S_i, \phi_j)$ by Eq. (7);
  M-step: Update the corresponding parameters
    $\tau^{(l+1)} = E[Z_{ij} = 1 \mid \pi_j^{(l)} S, \phi^{(l)}] = P(Z_{ij} = 1 \mid \pi_j^{(l)} S_i, \phi_j^{(l)})$ by Eq. (8);
    $\pi_j^{(l+1)} = \sum_{i=1}^{n} \frac{\tau_{ij}^{(l+1)}}{n}$ by Eq. (9), $\phi_j^{(l+1)}$ by Eq. (2.2);
    $\lambda_j^{(l+1)} = n_{k_j}^{(l+1)}$ by Eq. (11) ;
until $|\tau^{(l+1)} - \tau^{(l)}| < \epsilon$;

a fraction of Q function values. The M-step then sums up all these values and update the parameters. We use a $n \times g$ table storing the log-likelihood for each read calculated in E-step. The table has been randomly separated into x partitions, where each partition contains n/x reads. The latter is computed in x threads in parallel by using "IntStream" technique in Java. We summarize our workflow as shown in Fig. 2.

3 Results

We developed a PMM model and a Parallel algorithm (hence thereafter referred as PMMBin, Algorithm 1), to capture both long k-mer and short k-mer information in the DNA sequencing reads. We compared our methods to the competing methods that use long k-mers (i.e., AbundanceBin) only and short k-mers (i.e., MarkovBin) only.

3.1 Simulation Data Analysis

We used MetaSim [17], an open-source DNA sequencing reads simulation system, to generate six data sets, each with 10 million reads with 100 bases in length, which are "sequenced" from 10 randomly selected source species. We assigned the abundances of those species in the taxon profiles of MataSim following a normal distribution and used the empirical error model that was recommended for simulating Illumina reads. The ground truth of the reads abundances are shown in Fig. 3.

We compared the performance and running time of PMMBin and fPMM-Bin (derived from PMMBin by filtering out the inseparable DNA reads where

Table 2. The Accuracy (Acc.), Precision (Pre.), and adjusted Rand index (ARI) of PMMBin, fPMMBin, MarkovBin and AbundanceBin. The best performance results (excluding fPMMBin due to the added filtering procedure) are in bold face.

	PMMBin			AbundanceBin			MarkovBin			fPMMBin		
Data	Acc	Pre	ARI	Acc	Pre	ARI	Acc	Pre	ARI	Acc	Pre	ARI
1	0.77	0.85	**0.75**	0.56	0.59	0.14	0.59	0.81	0.56	0.96	0.86	0.95
2	0.70	0.76	**0.73**	0.42	0.65	0.12	0.44	0.74	0.44	0.93	0.78	0.92
3	0.84	0.85	**0.82**	0.52	0.63	0.15	0.55	0.82	0.53	0.92	0.86	0.91
4	0.75	0.80	**0.73**	0.50	0.66	0.24	0.68	0.79	0.66	0.90	0.82	0.90
5	0.63	0.91	0.54	0.43	0.57	0.13	0.90	0.74	**0.65**	0.98	0.88	0.81
6	0.66	0.84	0.51	0.56	0.67	0.22	0.99	0.63	**0.58**	0.91	0.87	0.76

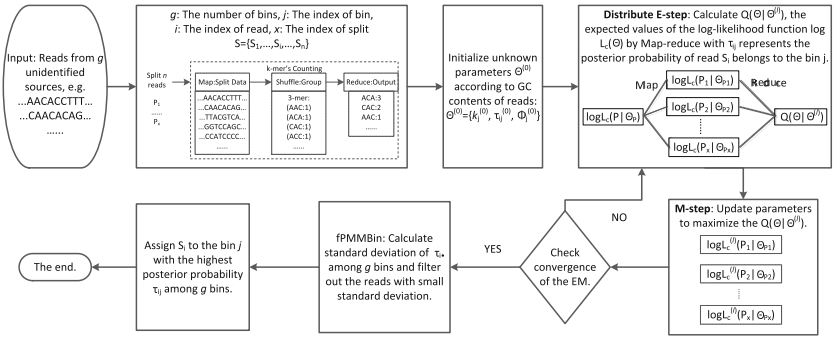
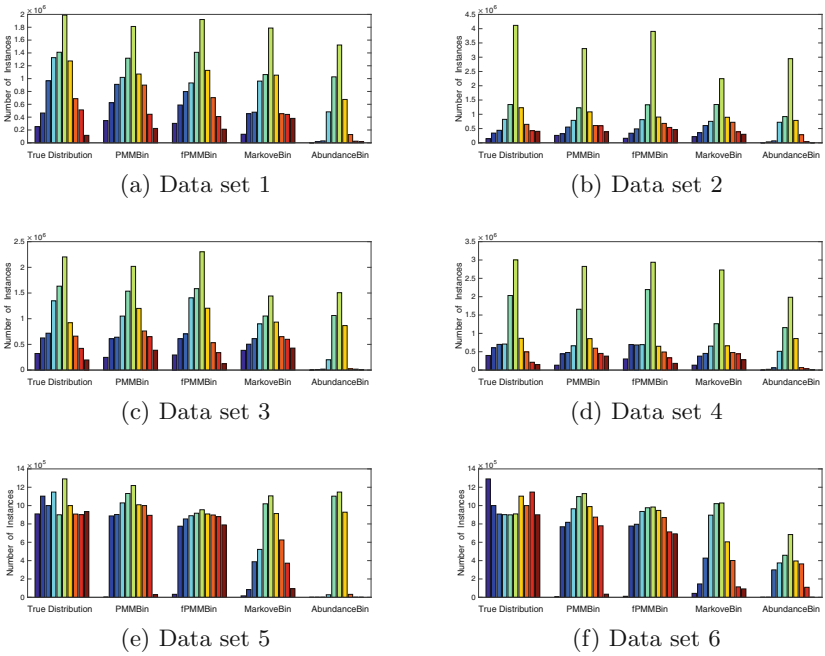


Fig. 2. A flowchart of the Parallel PMM algorithm implementation, where the dotted boxes represent a more efficient k-mer counting step to further speed up the algorithm.

standard deviation of $\tau_{ij}^{(l+1)}$ among clusters is less than 0.25) with that of AbundanceBin (long k-mers) [21] (version 1.01, February 2013) and MarkovBin (short k-mers) [16] (version 1.01, July 2013) in terms of accuracy, precision and adjusted Rand index (ARI) [5]. When calculating accuracy and precision, we consider a pair of reads to be positive if they are from the same source, negative otherwise. Let us denote N_P as the total number of the positive pairs, N_N as the total number of the negative pairs, N_{TP} (true positive) as the number of positive pairs that were assigned to the same bin, N_{TN} (true negative) as the number of negative pairs that were assigned to different bins. We define Accuracy as $\frac{N_{TP}}{N_P}$ and Precision as $\frac{N_{TN}}{N_N}$. To highlight the unique advantage of PMMBin in recovering the bin abundances with high variance, we designed a set of case-control experiments. Specifically, we used a bin size distribution with high variance to generate the data sets 1 to 4, and a true bin size distribution with low variance to generate the data sets 5 and 6.

From Table 2, PMMBin performs the best in the first 4 simulated data sets when compared with MarkovBin and AbundanceBin, but not in the last 2 data



(a) Data set 1 (b) Data set 2

(c) Data set 3 (d) Data set 4

(e) Data set 5 (f) Data set 6

Fig. 3. Comparison of reads binning performance in terms of recovering the true bin size distribution. Each panel corresponds to one data set and from left to right, bar plots represents: the True Distribution of bin size, the one estimated by our proposed PMM and fPMM approaches, the one estimated by MarkovBin approach and the one estimated by AbundanceBin approach.

sets, highlighting the unique capability of PMMBin in detecting bins of diverse sizes. Compared to PMMBin, fPMMBin enjoys much higher accuracy and ARI due to the removal of inseparable reads. Thus, the abundance variation information is duly captured by Poisson mixtures through extracting long k-mers while the base composition information is sufficiently captured by the mixture Markov models by extracting short k-mers. Therefore, our simulation studies strongly support the notion that short k-mers and long k-mers capture uncorrelated yet complementary feature information in the reads.

Figure 3 gives a more visually compelling comparison of the binning performance. PMMBin and fPMMBin successfully identified each of the 10 reads sources (species), represented by a "peak" for each source with negligible surrounding noises. Both MarkovBin and AbundanceBin miss a number of sources (peaks) albeit the former identities more sources than the latter. In data sets 1-4 when the bin sizes are truly diverse, the bin size distributions recovered by PMMBin and fPMMBin are much closer to the true distribution compared with MarkovBin and AbundanceBin. In data sets 5 and 6 when the bin sizes are more uniform, MarkovBin performs best whereas AbundanceBin capturing bin size variation performs the worst.

Table 3. Comparison of running time per iteration (100 million reads).

Data set	No Partition	4 Partitions	10 Partitions	40 Partitions
1	92.9 mins	41.7 mins	35.8 mins	6.2 mins
2	92.7 mins	41.3 mins	35.5 mins	5.8 mins
3	92.9 mins	41.6 mins	35.8 mins	6.1 mins
4	92.5 mins	41.4 mins	35.6 mins	5.9 mins
5	92.3 mins	41.1 mins	35.7 mins	5.8 mins
6	93.1 mins	41.7 mins	35.9 mins	6.2 mins

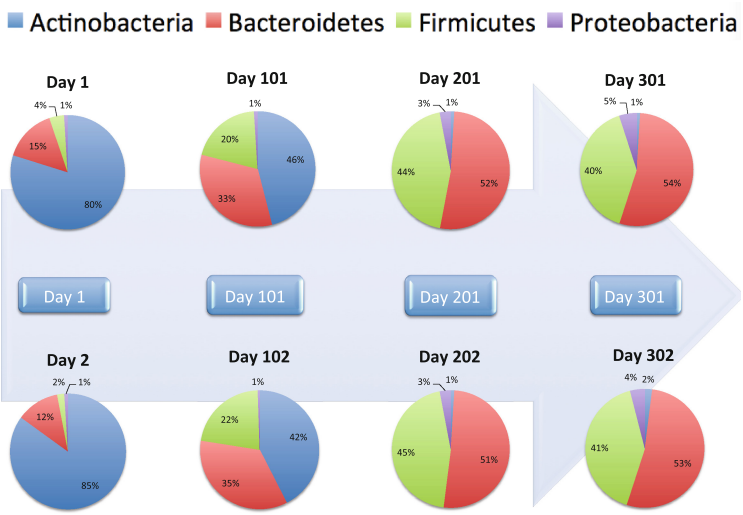


Fig. 4. The temporal changes of the individual's microbiome composition from Day 1 to Day 302.

We also compared running time of the parallelized PMM algorithm with the non-parallelized version. As shown in Algorithm 1, we split the calculation of expected log-likelihood into different number of partitions so that we calculate all partitions in parallel. We ran the parallel PMM algorithm on the 6 data sets (one hundred million reads) on a server (4x Twelve-Core AMD Opteron 2.6 GHz, 256 GB RAM). We compare the running time per iteration since different numbers of iterations are needed for different data sets. From Table 3, we observe a markedly faster running time of the parallelized PMM algorithm compared with the non-parallelized version without sacrificing the accuracy and precision.

Ideally, the running time of the parallelized algorithm per iteration can be reduced to $\frac{1}{x}$ of that of the non-parallelized algorithm, where x is the number of partitions. But it is not the case in reality as shown in Table 3. The reason is that we only parallelized the E-step since the E-step calculation dominates the entire computational complexity whereas M-step calculation is relatively trivial.

3.2 Real-World Data Analysis

We analyzed a human microbiome time course data set in which an individual's microbiome was sequenced daily over a period of one year [2]. We looked at the individual's microbiome data at eight days: day 1, day 2, day 101, day 102, day 201, day 202, day 301, and day 302, and we partitioned the reads from each metagenomic sample into four bins, i.e., Actinobacteria, Bacteroidetes, Firmicuts and Proteobacteria. From Fig. 4, it is evident that the individual's microbiomes are similarly between two consecutive days (per columns) whereas are radically different among distant days (per rows). It was also noted in [2] that the drastically changed microbiome at days 101–102 is due to the individual's trip abroad.

4 Conclusion

In this paper, we presented a novel probability model and a parallel algorithm to bin short DNA sequencing reads. Our original contributions lie in the systematic extraction and integration of both short and long k-mers information into the same probability model, and the parallel implementation of the optimization algorithm, which results in a vastly improved performance in terms of accuracy, precision and running time. Albeit the joint probability model was presented in the context of unsupervised reads binning, it is sufficiently flexible to be extended to solving supervised reads classification problems.

To further improve the running time, we will leverage efficient data structures for counting k-mers. Specifically, longer k-mers are sparse meaning that a majority of the k-mers are unique [15]. Thus the k-mers counting and hashing can be significantly accelerated by filtering out infrequent k-mers using a Bloom filter, and store frequent k-mers using a suffix tree in both memory and hard disk [10]. There are other existing k-mer counting approaches as well, such as in [13,22]. To this end, we will develop a versatile and scalable toolbox for facilitating data mining and machine learning of short DNA sequencing reads.

Acknowledgment. This research is partially supported by NSF grant CCF: 1451316 to D.Z.

References

1. Brady, A., Salzberg, S.L.: Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. Nat. Methods **6**(9), 673–676 (2009)
2. David, L.A., Materna, A.C., Friedman, J., Campos-Baptista, M.I., Blackburn, M.C., Perrotta, A., Erdman, S.E., Alm, E.J.: Host lifestyle affects human microbiota on daily timescales. Genome Biol. **15**(7), R89 (2014)
3. di Milano, U.C.S.: Poisson hidden markov models for time series of overdispersed insurance counts

4. Gerlach, W., Stoye, J.: Taxonomic classification of metagenomic shotgun sequences with CARMA3. Nucleic Acids Res. **39**(14), e91 (2011)
5. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)
6. Huson, D.H., Mitra, S., Ruscheweyh, H.-J., Weber, N., Schuster, S.C.: Integrative analysis of environmental sequences using MEGAN4. Genome Res. **21**(9), 1552–1560 (2011)
7. Kariin, S., Burge, C.: Dinucleotide relative abundance extremes: a genomic signature. Trends Genet. **11**(7), 283–290 (1995)
8. Karunanayake, C.: Multivariate Poisson Hidden Markov Models for Analysis of Spatial Counts. Canadian theses. University of Saskatchewan (Canada) (2007)
9. Kelley, D., Salzberg, S.: Clustering metagenomic sequences with interpolated Markov models. BMC Bioinform. **11**(1), 544 (2010)
10. Kurtz, S., Narechania, A., Stein, J.C., Ware, D.: A New Method to compute K-mer frequencies and its application to annotate large repetitive plant genomes. BMC Genomics **9**(1), 517 (2008)
11. Leroux, B.G., Puterman, M.L.: Maximum-Penalized-Likelihood estimation for independent and Markov-Dependent mixture models. Biometric **48**, 545–558 (1992)
12. Lu, J., Bushel, P.R.: Dynamic expression of 3' UTRs revealed by poisson hidden Markov modeling of RNA-Seq: implications in gene expression profiling. Gene **527**(2), 616–623 (2013)
13. Marçais, G., Kingsford, C.: A fast, lock-free approach for efficient parallel counting of occurrences of K-mers. Bioinform. **27**(6), 764–770 (2011)
14. Meinicke, P., Asshauer, K.P., Lingner, T.: Mixture models for analysis of the taxonomic composition of metagenomes. Bioinform. **27**(12), 1618–1624 (2011)
15. Melsted, P., Pritchard, J.K.: Efficient counting of K-mers in dna sequences using a bloom filter. BMC Bioinform. **12**(1), 333 (2011)
16. Nguyen, T.C., Zhu, D.: MarkovBin : an algorithm to cluster metagenomic reads using a mixture modeling of hierarchical distributions. In: Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics, p. 115. ACM (2013)
17. Richter, D.C., Ott, F., Auch, A.F., Schmid, R., Huson, D.H.: Metasim - a sequencing simulator for genomics and metagenomics. PLoS ONE **3**(10), e3373 (2008)
18. Salzberg, S.L., Delcher, A.L., Kasif, S., White, O.: Microbial gene identification using interpolated Markov models. Nucleic Acids Res. **26**(2), 544–548 (1998)
19. Wang, Y., Leung, H.C., Yiu, S.M., Chin, F.Y.: MetaCluster 4.0: a novel binning algorithm for NGS reads and huge number of species. J. Comput. Biol. J. Comput. Mol. Cell Biol. **19**(2), 241–249 (2012)
20. Wang, Y., Leung, H.C., Yiu, S.-M., Chin, F.Y.: Metacluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. Bioinform. **28**(18), i356–i362 (2012)
21. Wu, Y.-W., Ye, Y.: A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. J. Comput. Biol. **18**(3), 523–534 (2010)
22. Zhang, Q., Pell, J., Canino-Koning, R., Howe, A.C., Brown, C.T.: These are not the K-mers you are looking for: efficient online K-mer counting using a probabilistic data structure. PloS one **9**(7), e101271 (2014)

Springer

http://www.springer.com/978-3-319-38781-9

Bioinformatics Research and Applications
12th International Symposium, ISBRA 2016, Minsk, Belarus, June 5-8, 2016, Proceedings
Bourgeois, A.; Skums, P.; Wan, X.; Zelikovsky, A. (Eds.)
2016, XV, 348 p. 82 illus., Softcover
ISBN: 978-3-319-38781-9