MC-MinH: Metagenome Clustering using Minwise based Hashing

Zeehasham Rasheed * Huzefa Rangwala †

Abstract

Current bio-technologies allow sequencing of genomes from multiple organisms, that co-exist as communities within ecological environments. This collective genomic process (called metagenomics) has spurred the development of several computational tools for the quantification of abundance, diversity and role of different species within different communities. Unsupervised clustering algorithms (also called binning algorithms) have been developed to group similar metagenome sequences.

We have developed an algorithm called MC-MinH that uses the min-wise hashing approach, along with a greedy clustering algorithm to group 16S and whole metagenomic sequences. We represent unequal length sequences using contiguous subsequences or k-mers, and then approximate the computation of pairwise similarity using independent min-wise hashing. The performance of our algorithm is evaluated on several real and simulated metagenomic benchmarks. We demonstrate that our approach is computationally efficient and produces accurate clustering results when evaluated using external ground truth.

Keywords: min-wise hashing, sequence clustering, 16S and whole metagenomics

Website: `http://www.cs.gmu.edu/~mlbio/MC-MinH`

1 Background and Motivation

Metagenomics involves determination of the collective DNA of organisms co-existing as communities across various environments like sea, soil and human body [13, 14]. Metagenomics surpasses studying of individual genes and genomes, enabling scientists to study all of the genomes in a community as a whole. As an example, sequencing the microbial organisms within the human gut provides an understanding of the role played by microbes with regards to human health and disease [31].

However, sequencing technologies do not provide the whole genome of different co-existing organisms, but produce short contiguous subsequences called *reads* from random positions of the entire genome. Further, reads from different organisms are mixed together and a fundamental challenge in metagenome analysis is to stitch together these different sequence reads, using the overlap information to produce organism-specific genomes. This is referred to as the metagenome assembly problem (See Figure 1). Within a community, microbes (or organisms) vary in abundance, diversity, complexity, genome lengths and may have not been individually sequenced before. Besides, the current sequencing technologies produce large volume of sequence reads, and reads that may have varying error idiosyncracies [32]. As such, the metagenome assembly problem is complex and challenging [8].

Targeted metagenomics or 16S rRNA gene sequencing has been widely used for the analysis of genetic diversity within complex microbial communities. 16S sequences are marker genes, which exists in most microbial species but have variations in their sequences that allow them to be separated into different taxonomical groups [5]. Several metagenome analysis projects use sequencing of 16S genes as a first step in identifying the diversity within a sample.

Unsupervised clustering approaches (reviewed in Section 2) have been developed and used for the rapid analysis of large sets of whole and targeted 16S rRNA metagenomic sequences. Unsupervised grouping of sequences that belong to the same species is referred to as the "binning" problem. Successful binning (or grouping) of sequence reads assists in the metagenome assembly problem (See Figure 1), and also allows computation of different species diversity metrics. These approaches also reduce the complexity within computational workflows by determining cluster representatives that can be analyzed, instead of each individual sequence within a sample.

In this work, we develop a new, scalable metagenome sequence clustering algorithm called MC-MinH (Metagenome Clustering using Minwise based Hashing). Our approach is inspired from our previous work that used locality-sensitive hashing (LSH) to cluster 16S metagenomic sequences [26, 27]. MC-MinH uses min-wise hashing [2] along with a greedy clustering algorithm to group 16S as well as whole metagenomic sequences. This is achieved by representing unequal length sequences using contiguous subsequences and then approximating the computation of pairwise similarity (Jaccard similarity) using independent min-wise hashing.

We assess the performance of MC-MinH on two 16S simulated metagenomic samples, eight 16S environmental metagenomic samples [29] and fourteen simulated

*Department of Computer Science, George Mason University. Email: zrasheed@gmu.edu

†Department of Computer Science, George Mason University. Email: rangwala@cs.gmu.edu

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded from knowledgecenter.siam.org

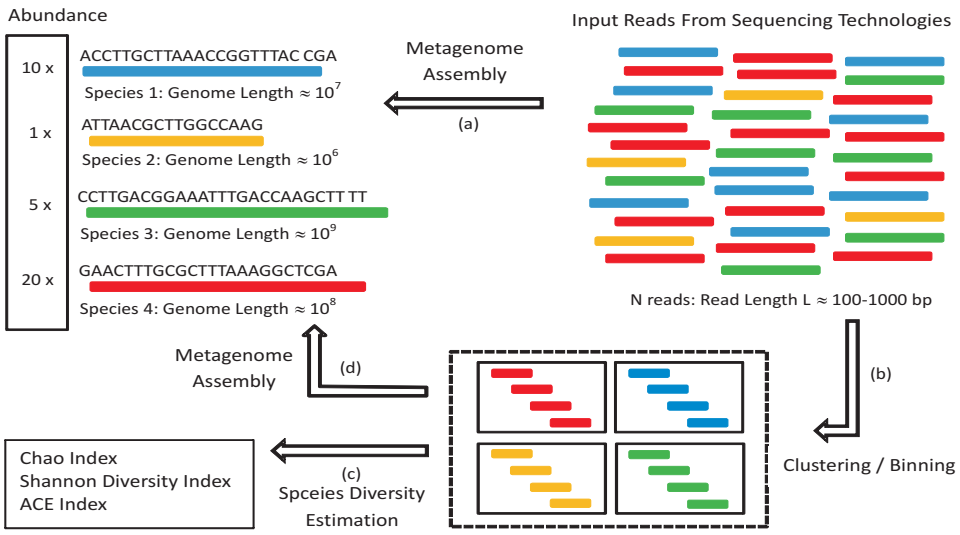

Figure 1: Metagenome Assembly and Clustering Problem.

The figure shows the process of metagenomic assembly and clustering. Samples are processed by sequencing technologies to produce metagenomic reads. (a) Metagenome assembly algorithms attempt to reproduce a complete organism specific genomes (b) Clustering algorithms are used to group reads in sequence specific groups to (c) determine species diversity or (d) aid in metagenome assembly.

and real whole metagenomic samples [9, 34]. Our results demonstrate the strength of MC-MinH in comparison to several other metagenome clustering algorithms when evaluated on the basis of clustering quality, computational run-time and pairwise sequence similarity of sequences within a cluster.

2 Related Work

Several computational algorithms have been developed for analyzing targeted and whole metagenome sequences [25, 18]. We review the approaches that are most relevant to the ideas discussed in this paper.

CD-HIT [23] and UCLUST [12] are general purpose sequence clustering algorithms that are widely used for clustering of 16S sequences. These algorithms follow a greedy approach and use pairwise sequence alignments (like edit distance) to find similar sequences. Speedup is achieved by using gapless, common matches of fixed lengths between sequence pairs called *seeds* to filter highly dissimilar sequence pairs.

On the other hand, DOTUR [28], MOTHUR [15] and ESPRIT [16] are exclusively designed for clustering 16S sequences. These methods use a pairwise distance matrix as input and then perform hierarchical clustering. For MOTHUR and DOTUR, pairwise distances are computed based on global alignment score between all pairs of sequences in the input set. An input parameter called distance cutoff is used to split the hierarchical representation (referred to as a dendrogram) at a specific level to produce different number of clusters. ESPRIT [16] is efficient in comparison to Mothur and DOTUR because it computes k-mer distance for each pair of input sequences, avoiding the expensive global alignment distance calculation. ESPRIT also implements several heuristics to reduce the number of sequence comparisons.

In our previous work, we developed MC-LSH [26], that utilizes an efficient locality sensitive based hashing function to approximate the pairwise sequence operations. Similarity among sequences is computed based on randomly chosen indices that essentially compresses the input sequence. In contrast, MC-MinH is designed to work for both, 16S and whole metagenomic sequences. Also, using a k-mer based representation and min-wise hashing allows us to efficiently cluster sequences that vary in length.

Recently, several algorithms have been developed for clustering whole metagenome sequence reads. Examples include TOSS [30], AbundanceBin [33], CompostBin [9], LikelyBin [20] and MetaCluster [34]. TOSS separates all unique k-mers into clusters and then merges these clusters based on k-mer repeat information. AbundanceBin models the sequencing reads as a mixture of Poisson distributions and uses an Expectation-Maximization (EM) algorithm to infer model parameters and the final clustering. CompostBin applies principal component analysis to project the data into an informative lower-dimensional space and then uses the normalized-cut clustering algorithm to group sequences into taxa-specific clusters. It also uses external information (supervised) of phylogenetic markers to assign clusters. LikelyBin uses an unsu-

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.
Downloaded from knowledgecenter.siam.org

Algorithm 1 MC-MinH algorithm for Metagenomic Clustering

Input: N sequences $S = \{s_1 \dots s_N\}$, each of variable length.
Parameters: k-mer size k, number of hash functions n, similarity threshold θ
Functions: I_s denotes k-mer size feature set for sequence s. $minHash\ (h(I))$ computes the min-wise hash value for I using hash function h .
Output: Cluster Assignments, indexed by different sequences $C[s_1 \dots s_N]$

1: Initialize array C, $\forall s_i \in S$, $C[s_i] = -1$.
2: $\forall s \in S$, compute k-mer feature set I_s
3: $\forall s \in S$, compute $minHash\ (h(I_s))$, where $h_i(x) = ((a_i x + b_i) \bmod p) \bmod m \mid x \in I_s$ and $i = 1, 2, \dots, n$
4: **repeat**
5: Choose $s_i \in S$, such that $C[s_i] == -1$ i.e., s_i is not assigned
6: Assign $C[s_i]$ to a **new** cluster label.
7: Remove s_i from S
8: **for** $\forall s_j \in S$ **do**
9: */* For all s_j in S that are not assigned, try to assign these sequences the same cluster label */*
10: **if** $\lfloor \frac{|minHash(I_{s_i}) \cap minHash(I_{s_j})|}{|minHash(I_{s_i}) \cup minHash(I_{s_j})|} \rfloor \geq \theta$ **then**
11: $C[s_j] \leftarrow C[s_i]$
/ s_i and s_j are in the same cluster */*
12: Remove s_j from S
13: **end if**
14: **end for**
15: **until** S is empty and all sequences are assigned.
16: return C

pervised, maximum-likelihood approach for clustering based on k-mer distributions of sequences; implemented using a MCMC approach. MetaCluster implements a two-phase (top-down separation and bottom-up merging) approach to cluster metagenomic fragments. The clusters are assigned on the basis of k-mer frequency and Spearman distance computation. MetaCluster was shown to produce the best metagenome clustering results in comparison to the approaches reviewed here [34]. We compare the performance of MC-MinH to state-of-the-art clustering algorithms on both, 16S and whole metagenomic benchmarks.

3 The MC-MinH Algorithm

In this section, we present our approach to cluster metagenome sequences using min-wise based hashing. The hashing approach allows us to reduce the complexity of exact pairwise string matching or multiple sequence alignment. We refer to our approach as MC-MinH (Metagenome Clustering using Minwise based Hashing). Min-wise hashing [3] efficiently computes approximate set of similarities within the context of document search. This idea is extensively applied for online content matching [1], documents and image retrieval [11], clustering and discovery of similar objects [10]. Recently Yang et. al. [35] proposed a Hadoop-based algorithm for the metagenomic clustering. They adapted the sketching technique [2] to identify pairwise sequence homologs and produced hierarchical taxonomic clustering results. However, their approach was benchmarked on 16S metagenomic sequence only. We first review the min-wise hashing approach and then describe our greedy clustering algorithm.

3.1 Min-Wise Hashing Algorithm Given a sequence represented as a set of k-mers (words/features), the similarity between two sequences can be defined as the Jaccard similarity between two corresponding sets of k-mers. If sequence s_1 has k-mer set denoted by I_{s_1} and sequence s_2 has k-mer set I_{s_2} then the Jaccard similarity is defined as:

$$sim(s_1, s_2) = \frac{|I_{s_1} \cap I_{s_2}|}{|I_{s_1} \cup I_{s_2}|}. \tag{3.1}$$

Now, we define the notion of min-wise independent families of permutations. Given $\mathcal{S}_n$ as the set of all independent permutations of $\lfloor n \rfloor = 1 \dots n$, we say that a family of permutations, $F \subseteq \mathcal{S}_n$ is min-wise-independent, if for any set $X \subseteq \lfloor n \rfloor$ and any $x \in X$, when π is chosen at random in F, we have

$$Pr(min\{\pi(X)\} = \pi(x)) = \frac{1}{|X|} \tag{3.2}$$

Thus, any element in the set X has an equal chance to become the minimum element of X under the permutation π, with equal probability [2]. Given a random permutation, every element has the same probability of being the smallest element in its subset.

Given the input n, k-mers (maximum value of $n = 4^k$) and given a min-wise independent permutation π chosen randomly from set F, the similarity between two sets (sequences s_1 and s_2) I_{s_1} and I_{s_2} is given by:

$$Pr(min\{\pi(I_{s_1})\} = min\{\pi(I_{s_2})\}) = \frac{|I_{s_1} \cap I_{s_2}|}{|I_{s_1} \cup I_{s_2}|} = sim(s_1, s_2) \tag{3.3}$$

The maximum value of $n = 4^k$ is the number of unique k-mers possible with the 4 nucleotides.

The probability of having the same minimum value within a permutation for two sets equals the Jaccard similarity between the two sets. As such, we can choose n independent random permutations $\pi_1, \pi_2, \dots, \pi_n$ and for each sequence s, we store the list:

$$\overline{s} = (min\{\pi_1(I_s)\}, min\{\pi_2(I_s)\}, \dots, min\{\pi_n(I_s)\}) \tag{3.4}$$

We can easily estimate the similarity between s_1 and s_2 by performing set intersection and set union of min-wise values in $\overline{s_1}$ and $\overline{s_2}$. The list $\overline{s_1}$ for s_1 is known as the fixed size sketch for representing the sequence.

3.2 Algorithm Details For given metagenome sequences represented by k-mer sets, we choose n

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.
Downloaded from knowledgecenter.siam.org

min-wise independent permutations, represented as $\pi_1, \pi_2, \ldots, \pi_n$. The problem with this approach is that it is not feasible to permute a large sequence set. Drawing random permutations is time consuming and practically inefficient. Fortunately, it is possible to simulate the effect of a random permutation by using universal hashing functions. This requires storing few hash values [22]. The standard universal hash function [4] is defined as

$$h_i(x) = (\ (a_i x + b_i) \bmod p\ ) \bmod m, \qquad i = 1, 2, \ldots, n \tag{3.5}$$

where m is the size of feature set, $p > m$ is a prime number and n is the number of hash functions. The parameters a_i and b_i are chosen uniformly from $\{0, 1, \ldots, p-1\}$. Instead of storing π_i, we now only need to store $2n$ numbers, a and b for each hash function.

For MC-MinH, instead of picking n random permutations, we pick n universal hashing functions $\{h_1, h_2, \ldots, h_n\}$ to approximate the random permutations. To compute the min-wise hash values for a given feature set I, we iterate over all k-mer features x and map them to their hash values $h_i(x)$. We then iterate over all the hash values to find their minimum, which will be the i^{th} min-wise value for that feature set. We formulate a $minHash$ function as the smallest element of a set I under ordering induced by the universal hashing function h, given by:

$$minHash\ (h(I)) = \arg\min_{x \in I} h(x) \tag{3.6}$$

Using the min-wise hashing property, the probability of hashing collision for two sets is equal to their Jaccard similarity. For metagenomic sequence clustering we are interested in finding those clusters which contain sequences that have similarity greater than some predefined threshold θ. Therefore, in MC-MinH algorithm, we formulate the probability of collision such that if $Pr[minHash\ (h(I_{s_1})) = minHash\ (h(I_{s_2}))] > \theta$, then s_1 and s_2 belong to the same cluster, otherwise a new cluster is created. The pseudo code of our MC-MinH algorithm is described in Algorithm 1. The input parameters include k-mer size k, number of hash functions n and similarity threshold θ. To compute the feature set I for given sequences (line 2), each sequence is converted to a set of fixed length subsequences of length k, called k-mers. This allows MC-MinH to handle variable length sequences. After the feature sets are computed, we use universal hash functions $h()$ defined in Equation 3.5 to compute hash value for each element in feature set I. At the end of this procedure, each sequence has n min-wise hash values, where n is the number of hash functions.

The MC-MinH algorithm follows a greedy, incremental procedure. After computing the min-wise values for each sequence, we begin by choosing the first sequence (or any one in the set S) and assign the sequence to the first cluster. Using the list representation (Equation 3.4), we can compute the Jaccard similarity between the unassigned sequences and the cluster representative. Sequences which have similarity greater than specified threshold θ are assigned to same cluster. We iterate through Steps 5-14 for all sequences in set S until cluster label is assigned to every sequence. When θ is set to 1, the MC-MinH algorithm will consider sequences to be similar if and only if all the min-wise values are identical in both sequences. Similarly, when θ is set to 0.95, sequences will go into the same cluster only if 95% of total min-wise values are similar in both sequences.

Table 1: Environmental DNA samples.

SID	Site	La°N, Lo°W	Dep	T	Reads
53R	Labrador seawater	58.300,-29.133	1,400	3.5	11218
55R	Oxygen minimum	58.300,-29.133	500	7.1	8680
112R	Lower deep water	50.400,-25.000	4,121	2.3	11132
115R	Oxygen minimum	50.400,-25.000	550	7.0	13441
137	Labrador seawater	60.900,-38.516	1,710	3.0	12259
138	Labrador seawater	60.900,-38.516	710	3.5	11554
FS312	Bag City	45.916,-129.983	1,529	31.2	52569
FS396	Marker 52	45.943,-129.985	1,537	24.4	73657

Description of the samples used in this paper. These samples are collected from North Atlantic Deep Water and Axial Seamount, Juan de Fuca Ridge. La is the latitude, Lo is the longitude, Dep is the depth in meters, T is the temperature in °C and Reads are the number of sequences in a sample.

4 Empirical Evaluation

4.1 Dataset Description We evaluate the performance of our algorithm on 16S and whole metagenomic benchmarks.

16S Simulated metagenomic samples: The simulated data contains 345,000 short sequences, generated from 43 known 16S rRNA gene fragments using the Roche GS20 system. This simulated data was originally used in a study by Huse et. al. [19].

16S Environmental samples: This dataset contains eight seawater samples taken from a study by Sogin et. al. [29]. For each of the samples, massively parallel DNA sequencing technology [24] (454/Roche) was used to efficiently increase the number of sequenced PCR amplicons. All the samples provide a global in-depth description of the diversity of microbes and their relative abundance in the ocean. Samples contain unequal length sequences with average sequence length of 60 bp. The description of these datasets are given in Table 1.

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.
Downloaded from knowledgecenter.siam.org

Table 2: Whole Metagenomic Sequence Reads.

SID	Species	Ratio	Taxonomic Difference	# Clu	# Reads
S1	Bacillus halodurans [0.44], Bacillus subtilis [0.44]	1:1	Species	2	49998
S2	Gluconobacter oxydans [0.61], Granulobacter bethesdensis [0.59]	1:1	Genus	2	49998
S3	Escherichia coli [0.51], Yersinia pestis [0.48]	1:1	Genus	2	49998
S4	Rhodopirellula baltica [0.55], Blastopirellula marina [0.57]	1:1	Genus	2	49998
S5	Bacillus anthracis [0.35], Listeria monocytogenes [0.38]	1:2	Family	2	49998
S6	Methanocaldococcus jannaschii [0.31], Methanococcus mariplaudis [0.33]	1:1	Family	2	49998
S7	Thermofilum pendens [0.58], Pyrobaculum aerophilum [0.51]	1:1	Family	2	49998
S8	Gluconobacter oxydans [0.61], Rhodospirillum rubrum [0.65]	1:1	Order	2	49998
S9	Gluconobacter oxydans [0.61], Granulobacter bethesdensis [0.59], Nitrobacter hamburgensis [0.62]	1:1:8	Family,Order	3	49996
S10	Escherichia coli [0.51], Pseudomonas putida [0.62], Bacillus anthracis [0.35]	1:1:8	Order, Phylum	3	49996
S11	Gluconobacter oxydans [0.61], Granulobacter bethesdensis [0.59], Nitrobacter hamburgensis [0.62], Rhodospirillum rubrum [0.65]	1:1:4:4	Family, Order	4	99998
S12	Escherichia coli [0.51], Pseudomonas putida [0.62], Thermofilum pendens [0.58], Pyrobaculum aerophilum [0.51], Bacillus anthracis [0.35], Bacillus subtilis [0.44]	1:1:1:1:2:14	Species, Order, Family, Phylum, Kingdom	6	99994
S13	Acinetobacter baumannii SDF, Pseudomonas entomophila L48	1:1	-	2	4000
S14	Ehrlichia ruminantium Gardel, Anaplasma centrale Israel, Neorickettsia sennetsu Miyayama	1:1:1	-	3	6000
R1	Glassy-winged sharpshooter endosymbionts	-	-	-	7137

Simulated (S) and Real (R) metagenomic shotgun datasets. Each dataset has a unique SID for reference. The GC content of each genome is written in [] brackets. S1-S12 are obtained from the work by Chatterji et al. [9] and S13-S14 are obtained from the work by Yang et. al [34].

Simulated and real whole metagenomic sequences: We also benchmark our algorithm on datasets derived from whole metagenome sequences. Within this dataset, we have 14 simulated samples with varying proportions of microbes and 1 real metagenomic sample. Sequence reads from multiple genomes are pooled to simulate the challenges of metagenomic sequencing i.e., varying number of species, relative abundance, phylogenetic diversity and varying composition of nucleotides between different genomes. These datasets are taken from a previous studies by Chatterji et al. [9] and Yang et al. [34]. All samples are described in Table 2. We also test our method on a publicly available metagenomic sample R1 that contains sequence reads obtained from gut of the glassy-winged sharpshooter *Homalodisca coagulata* (insect).

Table 3: Clustering Results on 16S Simulated dataset

	Dataset with 3% error		Dataset with 5% error	
Method	# Clu	W.Sim	# Clu	W.Sim
MC-MinH	**39**	97.70	37	95.18
MC-LSH	47	97.70	**41**	95.20
UCLUST	91	97.74	53	95.20
CD-HIT	108	97.74	47	95.20
ESPRIT	180	97.78	86	95.20
DOTUR	210	97.80	135	95.46
Mothur	214	97.80	138	95.46

These datasets contain sequence reads upto 3% and 5% errors with respect to reference 16S rRNA sequences. Number of clusters (# Clu) and weighted sequence similarity (W.Sim) in % are the performance metrics. The numbers in bold indicate that result produced by an algorithm is closer to the ground truth and performs better than other algorithms. Different number of clusters with similar weighted sequence similarity is due to the presence of single sequence clusters which are not included in calculating weighted sequence similarity.

4.2 Performance Metrics We evaluate the performance of our algorithm using different metrics and criteria. Specifically, we report the run time for different algorithms. Using ground truth from the simulated datasets i.e., taxonomic class labels for each sequence, we determine a weighted cluster accuracy. Each cluster is assigned to the class/genera based on the most frequent class in the cluster, and then the accuracy of this assignment is evaluated by computing the percent of correctly assigned sequences. The reported accuracy is averaged across all clusters, weighted by the number of sequences in each cluster. This is denoted by "W.Acc" in this paper. We also calculate the sequence similarity within the clusters for each clustering solution.

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.
Downloaded from knowledgecenter.siam.org

Table 4: Clustering Results on 16S Environmental sets.

Approach	Metric	53R	55R	112R	115R	137	138	F312(t)	F396(t)
MC-MinH	# Clu	1165	1077	1634	1156	1020	1042	1965	1340
	W.Sim	**96.90**	92.45	91.18	93.33	**95.86**	**93.10**	**91.25**	91.16
	Time (s)	2.5	2.1	3.3	3.0	2.7	2.5	3.4	2.4
MC-LSH	# Clu	1172	1199	1795	1205	1041	1072	1965	1363
	W.Sim	96.90	93.12	91.33	93.50	95.86	93.10	91.25	91.40
	Time (s)	161.0	183.0	317.0	188.0	172.0	175.0	112.0	101.0
UCLUST	# Clu	1062	992	1561	1071	900	923	1965	1346
	W.Sim	96.67	91.67	91.02	93.33	93.50	92.82	91.25	91.16
	Time (s)	2.0	2.0	2.0	2.0	2.0	2.0	1.0	1.0
CD-HIT	# Clu	824	716	1196	820	712	725	1626	945
	W.Sim	92.56	90.80	90.61	93.33	91.82	90.16	89.70	88.45
	Time (s)	3.6	3.1	3.9	3.8	3.2	3.1	3.8	2.5
ESPRIT	# Clu	940	859	1361	970	818	832	1936	1280
	W.Sim	93.12	91.35	90.88	93.33	91.82	90.16	91.20	89.50
	Time (s)	283.0	266.0	537.0	348.0	280.0	296.0	205.0	192.0
DOTUR	# Clu	1241	1258	1854	1279	1096	1121	2012	1381
	W.Sim	96.95	94.06	91.33	93.50	95.86	93.10	91.25	91.40
	Time (s)	5129.0	3511.0	5567.0	9237.0	6563.0	5618.0	1990.0	1457.0
Mothur	# Clu	1238	1256	1853	1278	1094	1119	2008	1372
	W.Sim	96.95	94.06	91.33	93.50	95.86	93.10	91.25	91.40
	Time (s)	10130.0	5940.0	12303.0	13501.0	12861.0	12310.0	2224.0	1583.0

For MC-MinH, all experiments are carried out with 15 k-mer and 50 hash functions. Number of clusters (# Clu), Weighted Similarity in % (W.Sim) and Running Time in seconds (Time) are the performance metrics. Similarity threshold of 95% is used for all the methods and Weighted Similarity is calculated separately to keep it consistent for all methods. Bold numbers indicate that MC-MinH performs better than other methods by producing similar Weighted Similarity with less number of clusters. F312(t) and F396(t) are trimmed down to smaller number of sequences because DOTUR and Mothur were unable to run on original dataset. MC-MinH works for both original and trimmed datasets.

Generally, sequence similarity is evaluated by finding the pairwise alignments (i.e., best arrangement of DNA nucleotides (characters) to identify regions of similarity between sequences). Global alignments attempt to align every character between the sequences, and local alignments find the best sub-regions of similar characters [17]. A good sequence clustering solution should produce sequences within a cluster that are similar to each other. We report only the average global sequence alignment similarity (weighted by number of sequences in a cluster). This quantity is referred to as "W.Sim" in this paper.

4.3 Hardware and Software Details The MC-MinH software is written in C language and runs on both windows and GNU linux operating systems. All experiments were performed on a single workstation, with Intel-i5 2.53 GHz processor and 6 GB memory. Comparative approaches were run on the same machine using binary files made available by the authors of these softwares.

4.4 Results and Discussion

16S Simulated dataset: We report in Table 3 the performance of different clustering algorithms for the 16S simulated set derived from 43 distinct genomes. We report results for reads that have less than 3% and 5% error. From Table 3 we observe that for 3% sequencing error, all algorithms overestimate the number of species except MC-MinH which produces 39 clusters (very close to 43). At 5% error, number of clusters produced by MC-LSH (41), MC-MinH (37) and CD-HIT (47) are closer to ground truth. We also report the weighted pairwise similarity within the clusters, and observe that MC-MinH shows promising weighted similarity results with less number of clusters.

16S Environmental dataset: Table 4 shows the run-time and similarity results of several clustering methods on 16S environmental samples. We can observe that MC-MinH outperforms other methods by producing similar weighted similarity (W.Sim) with less number of clusters. We see that time taken by MC-MinH, UCLUST and CD-HIT are close to each other

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.
Downloaded from knowledgecenter.siam.org

Table 5: Clustering results on Simulated and Real metagenomic dataset.

	MC-MinH				MetaCluster				LikelyBin			
SID	# Clu	W.Acc	W.Sim	Time	# Clu	W.Acc	W.Sim	Time	# Clu	W.Acc	W.Sim	Time
S1	7	86.98	**55.14**	**59.62s**	5	**87.46**	50.92	11m 25s	–	–	–	–
S2	6	**85.70**	**54.05**	**55.07s**	7	85.19	50.20	12m 10s	–	–	–	–
S3	8	**81.27**	**55.36**	**1m 02s**	7	80.84	51.85	12m 48s	–	–	–	–
S4	3	90.16	**57.40**	**1m 06s**	3	**92.34**	54.30	11m 33s	–	–	–	–
S5	4	**81.25**	**53.71**	**1m 01s**	5	79.69	50.71	12m 18s	–	–	–	–
S6	7	98.14	**56.92**	**50.62s**	8	**99.12**	56.67	13m 17s	–	–	–	–
S7	3	97.08	**56.60**	**54.13s**	4	**98.44**	55.18	12m 51s	–	–	–	–
S8	6	**93.65**	**55.78**	**56.25s**	5	92.29	53.05	13m 04s	–	–	–	–
S9	4	**86.20**	**56.62**	**1m 02s**	4	85.20	50.41	12m 28s	–	–	–	–
S10	5	97.44	**56.90**	**1m 09s**	7	**98.63**	54.66	13m 08s	–	–	–	–
S11	7	**88.52**	**53.10**	**2m 45s**	9	87.37	51.10	30m 25s	–	–	–	–
S12	10	95.32	**57.45**	**2m 57s**	12	**96.47**	53.94	32m 10s	–	–	–	–
S13	8	99.61	**58.96**	**3.81s**	4	**99.87**	53.33	4.34s	2	98.62	50.86	7m 52s
S14	6	**93.28**	**53.17**	**7.01s**	5	91.05	51.90	9.53s	2	90.83	51.64	11m 42s
R1	4	–	**55.80**	**4.90s**	3	–	50.45	12s	2	–	50.16	15m 08s

For MC-MinH, all experiments are carried out with 5 k-mer and 100 hash functions. Number of clusters (# Clu), weighted cluster accuracy in % (W.Acc), weighted sequence similarity (W.Sim) in % and Running Time in minutes or seconds (Time) are the performance metrics. The numbers in bold indicate that MC-MinH algorithm performs better than MetaCluster and LikelyBin. R1 represents the real metagenomic sample (7137 sequences). S1-S12 represent simulated metagenomic samples (each sample S1-S9 contains about 50,000 sequences of 1000 basepairs whereas each sample S11-12 contains about 100,000 sequences of 1000 basepairs). Sample S13 contains 4000 sequences of 1000 basepairs whereas sample S14 contains about 6000 sequences of 1000 basepairs. LikelyBin was unable to run on samples S1-S12 because of large number of sequences.

and MC-MinH outperforms MC-LSH, CD-HIT, ESPRIT, MOTHUR and DOTUR. Mothur and DOTUR also produce large number of clusters with expensive running times. The greater execution time of Mothur and DOTUR is due to the calculation of all pairwise distance computation for clustering.

Whole Metagenomic Sequences: Table 5 compares results for MC-MinH, MetaCluster and LikelyBin on simulated and real metagenomic samples. For datasets S1-S12, we have ground truth labels to compute the clustering accuracy based on species distribution. We see that MC-MinH outperforms MetaCluster in weighted sequence similarity across all samples. In terms of weighted cluster accuracy, the two approaches are statistically comparable. We observe that the MC-MinH algorithm is efficient in terms of execution time and is about 10 times faster than MetaCluster. Also, MC-MinH performs better than MetaCluster on real metagenomic sample R1 with respect to weighted sequence similarity.

We also evaluate MC-MinH, MetaCluster and LikelyBin algorithms on two smaller simulated whole metagenome datasets (S13 and S14), provided by MetaCluster [34]. We see that MC-MinH produces better results in terms of weighted sequence similarity. LikelyBin is expensive in terms of execution time even with the smaller datasets.

The number of clusters reported in these experiments are obtained after applying a threshold on the number of sequences in each cluster. Clusters having few sequences (< 50 on average) are filtered out in order to highlight prominent groups in the samples.

Parameter Sensitivity Analysis: We performed a series of experiments on 16S environmental dataset, varying the different parameters of the MC-MinH algorithm. The results are reported in supplementary paper which show the effect of varying k-mer size and varying number of hash functions on the number of clusters and the weighted sequence similarity, respectively. We observe that increasing k-mer size beyond 15 has no effect on the number of clusters. Also, the weighted sequence similarity is unchanged after increasing k-mer size beyond 15. We conclude that choosing a very large k-mer size has no effect on performance of MC-MinH algorithm.

Our results also show that increasing the number of hash functions beyond 50 has little change in the quality of clustering results. Using large number of hash functions actually increases the runtime of MC-MinH algorithm.

5 Significance and Impact: Species Richness Estimation.

Measuring species richness is an important objective for microbial ecologists. These estimates generate quantitative predictions of the number of co-existing species

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.
Downloaded from knowledgecenter.siam.org

Table 6: Species Richness Estimates for 16S Environmental dataset.

SID	# Clu	# Singleton	# Doubleton	Chao1 Index	Shannon Index	ACE Index
53R	1165	617	170	2276.3	4.4	2243.7
55R	1077	586	154	2182.8	4.6	2214.1
112R	1634	1015	223	3931.3	5.3	4202.7
115R	1156	649	181	2411.4	4.6	2455.8
137	1020	489	127	1992.2	4.8	1800.1
138	1042	477	168	1713.8	4.4	1760.3
FS312	1965	1572	255	8334.9	7.4	9868.7
FS396	1340	938	201	4292.3	6.7	5016.4

Measurement of different species richness metrics computed by MC-MinH algorithm for the samples characterized by Sogin et. al. [29].

and facilitate the process of comparing different microbial communities. It also offers a means to address several other challenges related to microbial diversity. For example, in the field of applied ecology and conservation biology, the number of species that remain in a community provides an ultimate evidence in the fight to preserve and restore disturbed communities.

We use MC-MinH clustering results to estimate different species richness metrics for 16S environmental dataset. Clusters (groups) represent taxonomy specific groups called operational taxonomical units (OTUs). Table 6 shows various metrics such as Chao1 index [6], Shannon Diversity index [21] and ACE (Abundance-based Coverage Estimator) index [7]. These estimates demonstrate the application use of MC-MinH algorithm for assessing the diversity of different environmental samples. The formulae for these estimations are stated in the supplementary paper.

6 Conclusion

We present a new, scalable metagenomic sequence clustering algorithm (MC-MinH) that utilizes min-wise hashing to quickly and accurately estimate pairwise sequence similarity. The MC-MinH algorithm works on both 16S and whole genome shotgun metagenomic samples and handles unequal length sequences. We evaluate MC-MinH on several metagenomic datasets and perform a comprehensive study of the different parameters and their impact on number of clusters, sequence similarity and computational time. We demonstrate that MC-MinH is computationally efficient in comparison to the state-of-the-art clustering algorithms, and also produces accurate clustering results with respect to external class labels. The code is written in C and is made available publicly under the GNU GPL license at the supplementary website. We plan to develop a parallel version of MC-MinH that can utilize distributed computing resources to further improve the throughput of complete analysis.

7 Acknowledgements

The work is supported by NSF Career Award IIS-1252318 and USDA Award 35205-17920 awarded to HR.

References

[1] A. Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences 1997*, SEQUENCES '97, pages 21–, Washington, DC, USA, 1997. IEEE Computer Society.

[2] Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60:327–336, 1998.

[3] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. In *Selected papers from the sixth international conference on World Wide Web*, pages 1157–1166, Essex, UK, 1997. Elsevier Science Publishers Ltd.

[4] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions (extended abstract). In *Proceedings of the ninth annual ACM symposium on Theory of computing*, STOC '77, pages 106–112, New York, NY, USA, 1977. ACM.

[5] Soumitesh Chakravorty, Danica Helb, Michele Burday, Nancy Connell, and David 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.

[6] Anne Chao. Nonparametric Estimation of the Number of Classes in a Population. *Scandinavian Journal of Statistics*, 11(4), 1984.

[7] Anne Chao and Shen M. Lee. Estimating the Number of Classes via Sample Coverage. *Journal of the American Statistical Association*, 87(417):210–217, 1992.

[8] Anveshi Charuvaka and Huzefa Rangwala. Evaluation of short read metagenomic assembly. *BMC Genomics*, 12.

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded from knowledgecenter.siam.org

[9] Sourav Chatterji, Ichitaro Yamazaki, Zhaojun Bai, and Jonathan A. Eisen. Compostbin: A dna composition-based algorithm for binning environmental shotgun reads. In *In: Research in Computational Molecular Biology, Proceedings*, pages 17–28, 2008.

[10] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: Finding a (thick) needle in a haystack. *CVPR*, 0:17–24, 2009.

[11] O. Chum, J. Philbin, and A. Zisserman. Near Duplicate Image Detection: min-Hash and tf-idf Weighting.

[12] Robert C. Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 2010.

[13] J. C. Venter et al. Environmental genome shotgun sequencing of the Sargasso Sea. *Science*, 304(5667):66, 2004.

[14] Junjie Qin et al. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, 464(7285):59–65, Mar 2010.

[15] Patrick D. Schloss et al. Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.*, 75(23):7537–7541, 2009.

[16] Sun Yijun et al. Esprit: estimating species richness using large collections of 16s rrna pyrosequences. *Nucleic Acids Research*, 2009.

[17] Xiaoqui Huang. On global sequence alignment. *Computer applications in the biosciences : CABIOS*, 10(3):227–235, 1994.

[18] Philip Hugenholtz and Gene W. Tyson. Microbiology: metagenomics. *Nature*, 455(7212):481–3, September 2008.

[19] Susan Huse, Julie Huber, Hilary Morrison, Mitchell Sogin, and David Welch. Accuracy and quality of massively parallel dna pyrosequencing. *Genome Biology*, 8(7):R143, 2007.

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

[21] Charles J. Krebs. *Ecological Methodology.* Harper & Row, New York., 1989.

[22] Ping Li, Anshumali Shrivastava, and Christian A. Konig. Gpu-based minwise hashing. In *Proceedings of the 21st international conference companion on World Wide Web*, WWW '12 Companion, pages 565–566, New York, NY, USA, 2012. ACM.

[23] Weizhong Li and Adam Godzik. CD-HIT: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[24] Marcel Margulies, Michael Egholm, and William E. Altman et al. Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–380, July 2005.

[25] Mihai Pop and Steven L Salzberg. Bioinformatics challenges of new sequencing technology. *Trends Genet*, 24(3):142–9, Mar 2008.

[26] Zeehasham Rasheed, Huzefa Rangwala, and Daniel Barbara. Efficient clustering of metagenomic sequences using locality sensitive hashing. In *SIAM International Conference in Data Mining*, pages 1023–1034, Anaheim, CA, 04 2012. SIAM.

[27] Zeehasham Rasheed, Huzefa Rangwala, and Daniel Barbara. LSH-Div:species diversity estimation using locality sensitive hashing. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Philadelphia, USA, 10 2012. IEEE.

[28] Patrick D. Schloss and Jo Handelsman. Introducing dotur, a computer program for defining operational taxonomic units and estimating species richness. *Appl. Environ. Microbiol.*, 71(3):1501–1506, 2005.

[29] Mitchell L. Sogin, Hilary G. Morrison, Julie A. Huber, David Mark Welch, Susan M. Huse, Phillip R. Neal, Jesus M. Arrieta, and Gerhard J. Herndl. Microbial diversity in the deep sea and the underexplored "rare biosphere". *Proceedings of the National Academy of Sciences*, 103(32):12115–12120, 2006.

[30] O. Tanaseichuk, J. Borneman, and T. Jiang. Separating metagenomic short reads into genomes via clustering. *Algorithms for Molecular Biology*, 7(1):27, 2012.

[31] P.J. Turnbaugh, R.E. Ley, M. Hamady, C.M. Fraser-Liggett, R. Knight, and J.I. Gordon. The human microbiome project. *Nature*, 449(7164):804–810, 2007.

[32] Gene W. Tyson and Philip Hugenholtz. Metagenomics. *Nature Reviews Microbiology*, Sep 2008.

[33] Y.W. Wu and Y. Ye. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. In *Research in Computational Molecular Biology*, pages 535–549. Springer, 2010.

[34] B. Yang, Y. Peng, H. Leung, SM Yiu, J. Qin, R. Li, and F.Y.L. Chin. Metacluster: unsupervised binning of environmental genomic fragments and taxonomic annotation. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 170–179. ACM, 2010.

[35] X. Yang, J. Zola, and S. Aluru. Parallel metagenomic sequence clustering via sketching and maximal quasi-clique enumeration on map-reduce clouds. In *Proc. IEEE Int. Parallel and Distributed Processing Symposium (IPDPS)*, pages 1223–1233, 2011.

Copyright © SIAM.
Unauthorized reproduction of this article is prohibited.

Downloaded from knowledgecenter.siam.org