Clustering Metagenome Sequences Using Canopies

Mohammad Arifur Rahman, Nathan LaPierre, Huzefa Rangwala and Daniel Barbara
Department of Computer Science, George Mason University
Fairfax, VA, United States
mrahma23@gmu.edu, nlapier2@gmu.edu, rangwala@cs.gmu.edu, dbarbara@gmu.edu

Abstract

Advances in biotechnologies has allowed for the collective sequencing of co-existing microbial communities (Metagenomics). These microbial communities are ubiquitous across various clinical and ecological environments. Metagenomics has spurred the development of several bioinformatics approaches for analyzing the diversity, abundance and functions of different organisms within these communities.

We present a fast and scalable clustering algorithm for analyzing large-scale metagenome sequence data. Our approach achieves efficiency by partitioning the large number of sequence reads into groups (called canopies) using locality sensitive hashing. These canopies are then refined by using state-of-the-art sequence clustering algorithms. This two-phase approach allows for the use of our developed canopy-clustering algorithm as a pre-processing phase for computationally expensive clustering algorithms.

We evaluate our clustering algorithm on synthetic and real world 16S and whole metagenome benchmarks. We demonstrate the ability of our proposed approach to determine meaningful OTU assignments and observe significant speedup with regards to run time when compared to three different clustering algorithms. We also make our source code publicly available on Github[1].

keywords: Clustering, Canopy, Metagenome, 16S, Biodiversity

1 Introduction

A large portion of the earth's biomass comprises trillions of tiny microorganisms with varying biodiversity. Communities of interacting microbes exists in several ecosystems ranging from the soil, ocean and human body[30]. Though, most of these microbes are beneficial to the host some are known to cause unwanted conditions and can be linked to cause of diseases in the host.

[1]https://github.com/mrahma23/LSH-Canopy

Metagenomics is the sequencing of the collective DNA of microbial organisms coexisting as communities. Analyzing metagenomes has provided an unprecedented opportunity to understand the diversity, role and function of these organisms within different clinical and ecological environments[27][9].

However, sequencing technologies do not deliver the complete genome of an organism (millions in length), but large number of short contiguous subsequences called *reads* (of length 75 to 500) in random order. Reads from different microbes are mixed together. Further, the composition of a community in terms of abundance and identity of microbial species is unknown[5]. Sequencing technologies also produce large datasets that range from Gigabytes (GB) to Terabytes (TB). 16S and 18S sequences contain repetitive marker genes that allow them to be separated into different pyhlogenetic groups[16].

In the past few years, several unsupervised clustering algorithms have been developed and used for the analysis of large-scale targeted and whole metagenome sequence reads (reviewed in Section 2). Clustering leads to assignment of similar sequences within groups known as *Operational Taxonomic Units (OTUs)*[12] which are ecologically consistent across the hosts[32].

In this paper we develop an efficient clustering algorithm based on the canopy framework[7] and a fast sequence distance metric based on locality sensitive hashing[4]. Sequences within the canopies are considered as an initial partition which can be further refined by applying expensive and accurate clustering algorithms. Our developed approach treats each canopies independently allowing for easy parallelizing (embarrassingly parallel).

We present experimental results on real and synthetic metagenome sequence benchmarks. We show that the use of Canopy Clustering (CC) in combination with UCLUST[20], SUMACLUST[14] and SWARM[23] leads to improved run time efficiency by 12.19, 21.09 and 18.61 times with accurate clustering results, respectively.

2 Related Work

In the past decade several sequence clustering methods have been developed and used for metagenome sequences. A comprehensive survey by Kopylova et. al.[21] benchmarks various approaches including UCLUST[20], SWARM[23], SUMACLUST[14] and MOTHUR[29].

CD-HIT[22] is a general purpose sequence clustering algorithm that follows an incremental, greedy approach. CD-HIT uses pairwise sequence alignment to find similar sequences. UCLUST[20] is similar to CD-HIT but achieves a significant speedup over CD-HIT by using seeds (fixed length gapless subsequences) for performing pairwise sequence comparisons. MC-LSH[11] utilizes an efficient locality sensitive based hashing function to approximate the pairwise sequence similarity. MC-MinH[10] uses min-wise[1] hashing along with greedy clustering to group 16S and whole metagenome sequences. Mash[25], uses MinHash locality sensitive hashing to reduce large sequences to a representative sketch and estimate pairwise distances. Other methods for clustering sequence reads include TOSS[8], AbundanceBin[31] and CompostBin[2]. All unique kmers are first grouped in TOSS and then clusters are merged based on kmer repetitions. In AbundanceBin, reads are modeled as a mixture of Poisson distributions. Expectation Maximization (EM) algorithm is used to infer model parameters for the final clustering. Principal component analysis is used within CompostBin to project the data into a lower dimensional space followed with a graph partitioning approach. MOTHUR[29] uses a pairwise distance matrix as input and performs hierarchical clustering.

SWARM[23] uses exhaustive single-linkage clustering based on optimal sequence alignment. Sequences that are less than a certain distance from any other sequences are clustered together. SWARM attempts to reduce the impact of clustering parameters on the resulting OTUs by avoiding arbitrary threshold parameters and input sequence ordering dependencies. SWARM builds an initial set of OTUs by iteratively agglomerating similar amplicons. The amplicon abundance values are used to reveal OTUs internal structures and break them into sub-OTUs.

SUMACLUST[14] is similar to UCLUST and uses a greedy strategy to incrementally construct clusters by comparing an abundance-ordered list of input sequences against the representative set of already-chosen sequences. UCLUST, SWARM and SUMACLUST are considered to be state-of-the-art metagenome sequence clustering methods by a benchmarking study[21]. We compare the performance of our developed method with these three approaches.

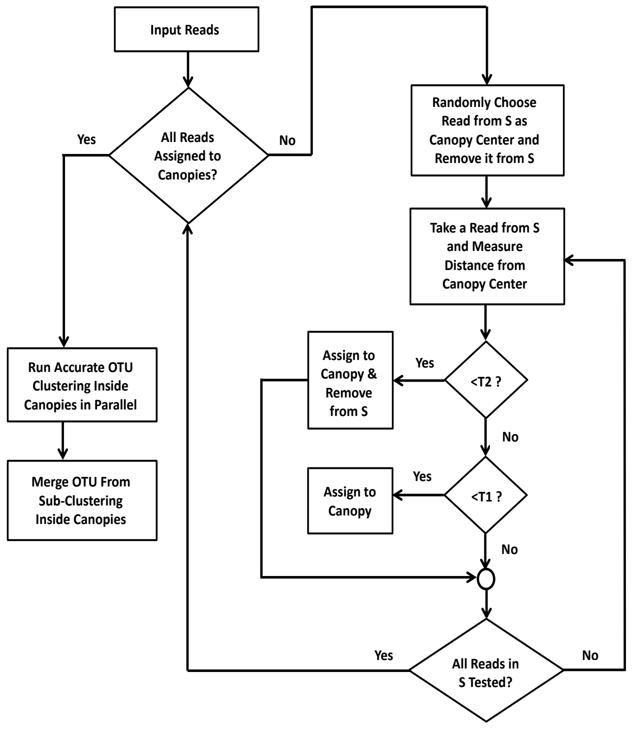

Input Reads
All Reads Assigned to Canopies?
Yes
No
Randomly Choose Read from S as Canopy Center and Remove it from S
Take a Read from S and Measure Distance from Canopy Center
Run Accurate OTU Clustering Inside Canopies in Parallel
Merge OTU From Sub-Clustering Inside Canopies
Assign to Canopy & Remove from S
Yes
<T2 ?
No
Assign to Canopy
Yes
<T1 ?
No
Yes
All Reads in S Tested?
No

Figure 1: Workflow of Canopy Clustering for Large Scale Metagenome Data

3 Methodology

3.1 Overview

Figure 1 provides an overview of our proposed canopy clustering algorithm. Sequence reads are represented with *kmers*, defined as contiguous subsequence of length k. Kmers are then hashed according to LSH principle and each sequence read is assigned to at least one canopy. More accurate clustering methods are used to sub-cluster each canopy in parallel. Canopy clustering significantly reduces pairwise distance computations. We provide more details for each of these steps below.

3.2 Canopy Clustering

Canopy Clustering[7] is an efficient approximate clustering algorithm often used as pre-processing step for other accurate and expensive clustering methods. It is intended to speed up the clustering operations for large data sets by reducing number of pairwise distance calculations, where standard clustering algorithms may be impractical due to run time and memory requirements.

Canopy clustering uses two distance thresholds, (i) *soft* threshold, $T1$ and (ii) *tight* threshold, $T2$. If a data instance p_1 is within the soft distance threshold $T1$ with centroid C_a then p_1 will reside in same canopy as C_a. However, p_1 may belong to other canopies assuming that it has only met soft threshold and it's best match is yet to be found. Thus one data point may belong to multiple canopies. However, if data point p_1 is within the tight distance threshold $T2$ with centroid C_a then canopy clustering assigns p_1 to the canopy with centroid C_a and stops assigning p_1 to any other canopy. Initially, canopy centroids are selected randomly until all data points are assigned to at least one canopy.

3.3 Locality Sensitive Hashing

Canopies are intended to reduce the total pairwise distance calculations. As such, canopy clustering should be efficient, which requires a fast and approximate distance measure. Locality Sensitive Hashing (LSH)[4] provides a solution for the approximate or exact near neighbors search problem. We construct the family of LSH functions with bit sampling[6]. Given an input vector v and a random hyperplane defined by r; hash function $h(v) = \{0, 1\}$ based on $sgn(v \cdot r) = \pm 1$ that indicates on which side of the hyperplane v lies. For any i^{th} bit, h_i and for any two data instances x, y, the probability that x and y *agree* on the i^{th} positions of their respective d-length binary vector is

$$P[h_i(x) = h_i(y)] = 1 - \frac{f(x, y)}{d} \tag{1}$$

where $f(x, y)$ is the hamming distance and d is the number of bits. The random projection and hamming distance calculation is computationally efficient making it suitable for fast partitioning of large volume of sequence reads.

3.4 Sub-Clustering Inside Canopies

Each of the canopy-based partitions are further clustered in parallel and independendtly with expensive (but accurate) clustering methods. We use UCLUST[20], SUMACLUST[14] or SWARM[23] as accurate and expensive sub-clustering methods.

3.5 Merging Results from Canopies

Each cluster (OTU) is represented by the Longest Common Subsequence of all member sequences. The final step of our proposed framework is to merge the OTU representations generated by the canopies. According to Canopy cluster algorithm a single data point may belong to multiple canopies as long as the soft threshold is met. As a result similar OTU representations may appear from multiple canopies. To eliminate redundancy we run UCLUST on the OTU representations.

For notation purposes, the Canopy clustering (CC) approach integrated with UCLUST, SUMACLUST and SWARM is denoted by CC_{UCLUST}, $\text{CC}_{SUMACLUST}$ and CC_{SWARM}, respectively.

4 Experimental Results

4.1 Dataset Description

To evaluate the performance of our proposed approach we use previously published synthetic and real world 16S, 18S and metagenome sequence benchmarks. Key statistics regarding these datasets are presented in Table 1.

Table 1: **Dataset Statistics**

Datasets	**Type**	**# of Reads**	**# of Samples**	**Platform**
Bokulich$_2$ [13]	M	6,938,836	4	H
Bokulich$_3$ [13]	M	3,594,237	4	H
Bokulich$_6$ [13]	M	250,903	1	H
Canadian Soil [24]	R	2,966,053	13	H
Body Sites [17]	R	886,630	602	G
Global Soil [28]	R	9,252,764	57	H
Liver Cirrhosis [3]	R	6,117,828,130	232	H

Table shows information about dataset used in this study. M, R, H and G represent Mock, Real World, HiSeq and GS-FLX, respectively.

Table 2: **Parameter Settings**

Datasets	Total # of Reads	# of Reads in Sampled Data	Parameters		
			d	$T1$	$T2$
Bokulich$_2$	6,938,836	693,884	47	0.48	0.36
Bokulich$_3$	3,594,237	359,428	31	0.43	0.34
Bokulich$_6$	250,903	25,090	17	0.37	0.21
Canadian Soil	2,966,053	296,605	29	0.46	0.37
Body Sites	886,630	88,663	22	0.39	0.22
Global Soil	9,252,764	925,276	59	0.49	0.38
Liver Cirrhosis	3,000,000	300,000	48	0.42	0.35
Liver Cirrhosis	30,000,000	3,000,000	67	0.51	0.39

4.1.1 Synthetic Datasets:

The synthetic datasets are from Bokulich et al.[13] and available at QIIME database[2] under their respective ID's. Since, these are simulated datasets the taxonomic profile of microbial organisms within them are known. We have used three well known simulated datasets Bokulich$_2$, Bokulich$_3$ and Bokulich$_6$. Bokulich$_2$, a simulated 16S rRNA gene microbial community dataset, was prepared using the Illumina TruSeq v2 paired–end library preparation kit. It contains 19 taxonomic families, 19 genera, 22 species and 22 strains in total. Bokulich$_3$ is similar to Bokulich$_2$ except that it was

[2]http://qiime.org/home_static/dataFiles.html

Table 3: **Runtime Comparison (in minutes)**

Datasets			**Methods**								
Type	Title	# of Reads	UCLUST	CC_{UCLUST}	Speed Up	SUMACLUST	$CC_{SUMACLUST}$	Speed Up	SWARM	CC_{SWARM}	Speed Up
Synthetic	$Bokulich_2$	6,938,836	12.71	**3.08**	**4.13x**	114.53	**13.89**	**8.24x**	128.12	**17.24**	**7.43x**
	$Bokulich_3$	3,594,237	10.43	**3.61**	**2.89x**	27.73	**5.11**	**5.43x**	18.37	**3.39**	**5.41x**
	$Bokulich_6$	250,903	4.47	**2.09**	**2.14x**	5.61	**2.07**	**2.71x**	4.17	**1.29**	**3.21x**
Real World	Body Sites	886,630	9.46	**3.03**	**3.12x**	18.42	**6.02**	**3.06x**	16.37	**5.83**	**2.81x**
	Canadian Soil	2,966,053	13.65	**4.17**	**3.27x**	363.96	**56.61**	**6.43x**	117.53	**20.84**	**5.64x**
	Global Soil	9,252,764	108.21	**18.75**	**5.77x**	510.92	**45.37**	**11.26x**	289.51	**34.84**	**8.31x**
	Liver Cirrhosis Metagenome	3,000,000	14.57	**4.03**	**3.61x**	46.62	**9.02**	**5.17x**	41.37	**8.75**	**4.73x**
	Liver Cirrhosis Metagenome	30,000,000	22.41h	**1.84h**	**12.19x**	46.62h	**2.21h**	**21.09x**	37.43h	**2.01h**	**18.61x**

prepared with the TruSeq v1 paired-end library kit at Illumina Cambridge. $Bokulich_6$ is 16S rRNA dataset, sequenced at Washington University School of Medicine and contains evenly distributed microbial communities. This dataset contains 13 taxonomic families, 23 genera, 44 species and 48 strains in total.

4.1.2 Real World Datasets

Canadian Soil: The Canadian Soil dataset[3] contains genomic data of soil samples collected from across Canada spanning multiple biomes and ecozones.

Body Sites: This dataset contains bacterial communities from up to 27 different body sites in healthy adults. A collection of 602 samples acquired from different body sites of human subjects are provided with meta-data.

Global Soil: The global soil data was taken from Ramirez et al.[28] which is a study of the below-ground diversity in New York City's Central Park.

Liver Cirrhosis: The Liver Cirrhosis dataset was taken from the study by Qin et al.[3]. This is a whole gut microbiome wide association study of stool samples from 98 liver cirrhosis patients and 83 healthy controls. Because of the high volume of sequence reads in this dataset, we used two random samples of 3 million and 30 million sequence reads from the original dataset.

4.2 Evaluation Metrics

We evaluate the performance of our developed clustering approach using the following commonly used metrics that are used for the assessment of (i) outputs from clustering algorithms, (ii) biodiversity within metagenome samples and (iii) computational run time.

4.2.1 Faiths Phylogenetic Diversity Metric (PD)

Faiths phylogenetic diversity[18] combines all the branch lengths of phylogenetic tree as a measure of diversity. If a new OTU is found and it is closely related to another OTU in the sample, it will contribute to a small increase to the PD score. However, if a new OTU from different lineage is found then it will contribute to a large increase in the PD score.

[3]http://www.cm2bl.org/

4.2.2 Shannon Entropy

Shannon-Wiener diversity index is defined as:

$$H = -\sum_{i=1}^{s} (p_i \log_2 p_i) \tag{2}$$

where s is the number of OTUs and p_i is the proportion of the community represented by OTU i. The Shannon index increases as both the richness and evenness of the community increase.

4.2.3 Simpson's Index

Simpsons index is defined as $1 - \mathit{dominance}$ or

$$1 - \sum p_i^2 \tag{3}$$

where where p_i is the proportion of the community represented by OTU i. Simpsons index is based on the probability that any two individuals drawn at random from an infinitely large community belong to the same species.

4.2.4 F-Score

In case of synthetic datasets, expected taxonomic composition is known (*ground truth*). False-positive (FP) refers to the number of taxonomy that was found in observed but not expected, false-negative (FN) refers to the number of taxonomy that exists in expected but not observed, and true-positive (TP) refers to the number of taxonomy exists in both observed and expected. The following definitions were used:

$$precision = \frac{TP}{(TP + FP)} \tag{4}$$

$$recall = \frac{TP}{(TP + FN)} \tag{5}$$

$$FScore = \frac{2 \times precision \times recall}{(precision + recall)} \tag{6}$$

4.2.5 Pearson Coefficient Correlation (ρ-value)

After getting OTUs from a clustering method, we create a taxonomic profile at the Genus level. Pearsons correlation coefficient was computed to measure the relatedness of taxonomic assignment between a pair of tools. Values range between -1 and 1, with -1 indicating a negative correlation, 0 indicating no correlation, and 1 indicating a positive correlation or strong relationship.

4.3 Parameter Settings

For kmers, the value of parameter k was set to 4. The parameters for bit length of LSH (d), canopy's soft ($T1$) and hard ($T2$) thresholds were set by performing a grid search and validation. For validation purposes, 10% of the data was randomly sampled. First, the parameter for LSH bit length d was estimated by fixing the soft ($T1$) and tight ($T2$) thresholds to 0.6 and 0.4, respectively. d was found to be proportional to the size of the input dataset. Once bit length for LSH (d) was estimated we tuned values for $T1$ and $T2$ by decreasing $T2$ from 0.4 (*relaxed*) to 0.1 (*strict*) in steps of 0.01 and varying $T1$ from 0.6 to 0.1 in steps of 0.01. The parameters obtained on the validation sets for different benchmarks are shown in Table 2.

4.4 Hardware and Software for Experiments

We performed all the experiments on computers with Intel 5^{th} generation Core i7, 2.70 GHz 64bit processor with 8 core CPUs and 12 GB memory. Sequence identity threshold for sub-clustering methods were set to default 97%. For implementation we used Python 2.7.12 and QIIME[15] version 1.9.0 (for diversity estimation). Taxonomy for reported OTUs was assigned using the RDP Classifier[33] against the 97% representative databases for Greengenes[19] and Silva[26]. We used the PyNast[4] open source sequence aligner for aligning clustered output.

5 Result

5.1 Runtime Comparison

Table 3 shows the run time in minutes for UCLUST, SUMACLUST, SWARM and their respective versions with our proposed Canopy clustering pipeline. For the 30M sequence dataset, CC outperforms UCLUST, SUMACLUST and SWARM by 12.19, 21.09 and 18.61 times, respectively. For the Global Soil dataset CC outperforms UCLUST, SUMACLUST and SWARM by 5.77, 11.26 and 8.31 times, respectively. The highest gains in run time were found for the larger datasets demonstrating that CC scales well with large scale data.

[4]http://biocore.github.io/pynast/

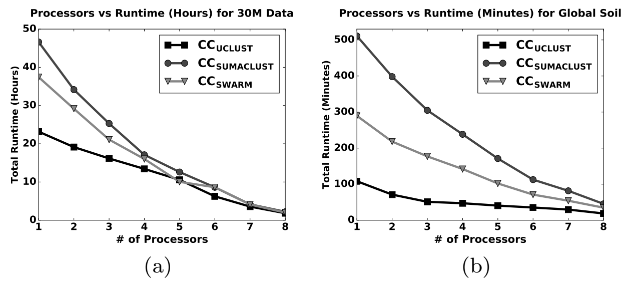


Figure 2: Effect of number of processors on Runtime of CC_{UCLUST}, $CC_{SUMACLUST}$ and CC_{SWARM} for 30M sampled dataset and Global Soil dataset, two of the largest datasets used in this study.

5.2 Effect of Varying Number of Processors

Figures 2a-2b show the runtime of CC_{UCLUST}, $CC_{SUMACLUST}$ and CC_{SWARM} on the two large benchmarks studied here. We observe that increasing number of processors reduces the total run time. Significant reductions in run time were observed for $CC_{SUMACLUST}$ and CC_{SWARM} as compared to CC_{UCLUST}.

5.3 Clustering Correctness

Table 4 shows the performance of UCLUST, SUMACLUST and SWARM in comparison to CC clustering versions. Table 4 compares F-scores and Pearson Correlation Coefficient (ρ). F-scores are only available for synthetic benchmarks since taxonomic profile for them is known. Correlation values were generated based on taxonomic profiles at the Genus level, after obtaining final cluster output. We observe that F-scores obtained from a clustering method and its corresponding Canopy clustering version are very similar and in some cases better. For Bokulich$_3$ benchmark we observed higher F-scores for all *CC* based methods. Finally for Bokulich$_6$ benchmark, the F-scores of CC_{UCLUST} and CC_{SWARM} were improved comparing to UCLUST and SWARM, respectively. For all benchmarks we observe strong correlations between taxonomic profiles at genus level.

Table 4: **Performance Comparison [F-Score and Pearson Correlation Coefficient (ρ)]**

Methods	**Comparison Metric**	**Datasets**							
		Synthetic			*Real World*				
		Bokulich$_2$	Bokulich$_3$	Bokulich$_6$	Body Sites	Canadian Soil	Global Soil	Liver Cirrhosis Metagenome (Sampled 3M)	Liver Cirrhosis Metagenome (Sampled 30M)
UCLUST	F-Measure	0.39	0.40	0.51	N/A	N/A	N/A	N/A	N/A
CC$_{UCLUST}$	F-Measure	**0.39**	**0.41**	**0.52**	N/A	N/A	N/A	N/A	N/A
	(ρ) with Respect to UCLUST	**0.9807**	**0.9753**	**0.9831**	**0.9682**	**0.8419**	**0.9824**	**0.9216**	**0.9072**
SUMACLUST	F-Measure	0.40	0.41	0.51	N/A	N/A	N/A	N/A	N/A
CC$_{SUMACLUST}$	F-Measure	**0.41**	**0.42**	**0.51**	N/A	N/A	N/A	N/A	N/A
	(ρ) with Respect to SUMACLUST	**0.9709**	**0.9813**	**0.9538**	**0.9518**	**0.7643**	**0.8714**	**0.9281**	**0.8614**
SWARM	F-Measure	0.46	0.48	0.55	N/A	N/A	N/A	N/A	N/A
CC$_{SWARM}$	F-Measure	**0.46**	**0.49**	**0.56**	N/A	N/A	N/A	N/A	N/A
	(ρ) with Respect to SWARM	**0.9817**	**0.97861**	**0.9251**	**0.9648**	**0.7581**	**0.9143**	**0.9263**	**0.8533**

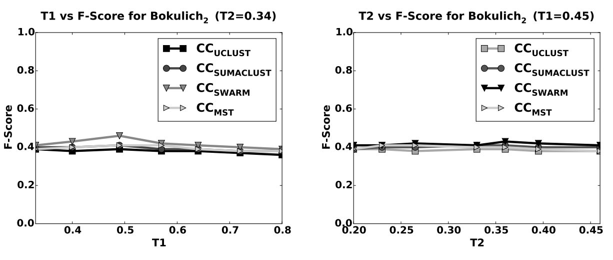
Table shows values of F-scores and Pearson Correlation Coefficient (ρ-value) of UCLUST, SUMACLUST, SWARM and their respective versions with Canopy clustering. F-measures is only available for synthetic datasets. ρ-value was calculated based on the taxonomy profiles at Genus level. Higher F-scores and ρ values are represented with bold letters.

5.4 Biodiversity Estimation

Table 5 shows the Faiths phylogenetic diversity metric (PD), Shannon and Simpson index after clustering with different methods. These metrics are popular Alpha Diversity metrics that measure species diversity in sites or habitats at a local scale. These metric values are generated per sample basis. Table 5 shows ranges of metric values in [$minimum-maximum$] format for the different methods with and without Canopy clustering. From Table 5, we observe that Canopy clustering based methods produce similar ranges of values as their naive counterparts.

5.5 Sensitivity Analysis (Varying T1 and T2)

Figure 3a-3b shows the effect of varying $T1$ ($T2$ fixed at 0.34) and $T2$ ($T1$ fixed at 0.45) on F-scores for the Bokulich$_2$ dataset. Reducing $T1$ leads to comparatively *strict* soft-threshold which will reduce repetitions of instances in multiple canopies. For a fixed $T2$ this implies that lower $T1$ will yield better canopy assignment. From Figure 3a we see that when $T1$'s range is in 0.4 to 0.6 our proposed approach provides better F-Scores. On the other hand, increasing $T2$ leads to comparatively *relaxed* tight-threshold for canopies. As a result instances will be prematurely assigned to canopies without waiting for best match. From Figure 3b we note that when $T2$ is in between 0.25 to 0.35, our proposed approach achieves better F-Scores. Lowering $T2$ may lead to higher run time since instances will continue to reappear until the $T2$ threshold is satisfied.



(a) (b)

Figure 3: 3a and 3b show effect of varying $T1$ and $T2$ on F-scores for the largest synthetic dataset Bokulich$_2$

6 Conclusion

We developed a greedy clustering algorithm that scales to large metagenome sequence datasets. The strengths of our proposed approach is that it can be integrated as a pre-processing step with state-of-the-art expensive but accurate clustering algorithms allowing them to operate efficiently on real world datasets. Our approach achieves efficiency by partitioning the large number of sequence reads into groups (called canopies) using locality sensitive hashing. This initial approximate assignment of sequence reads to canopies is then refined. We demonstrate that our approach provides similar outcome in terms of biodiversity metrics, ground truth and taxonomic correlation with corresponding expensive clustering methods. We show significant reduction in run times in comparison to previously developed sequence clustering algorithms, especially on datasets with millions of sequence reads.

References

[1] Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3):630–659, 2000.

[2] Sourav Chatterji, Ichitaro Yamazaki, Zhaojun Bai, and Jonathan A Eisen. Compostbin: A dna composition-based algorithm for binning

Table 5: **Biodiversity Comparison [Faiths phylogenetic diversity metric (PD), Shannon and Simpson]**

Methods	**Comparison Metric**	**Datasets**						
		Synthetic			*Real World*			
		Bokulich$_2$	Bokulich$_3$	Bokulich$_6$	Body Sites	Canadian Soil	Global Soil	Liver Cirrhosis Metagenome (Sampled 3M)
UCLUST	PD Range	[171.95 – 221.85]	[186.90 – 212.84]	[104.51 – 104.51]	[1.46 – 46.79]	[0.30 – 1352.73]	[2.98 – 3.29]	[3.14 – 51.47]
	Shannon Range	[2.52 – 3.51]	[2.43 – 3.54]	[5.87 – 5.87]	[0.29 – 7.67]	[2.32 – 7.86]	[1.84 – 8.30]	[0.27 – 5.77]
	Simpson Range	[0.55 – 0.75]	[0.55 – 0.76]	[0.96 – 0.96]	[0.049 – 0.98]	[0.80 – 0.99]	[0.0 – 0.98]	[0.02 – 0.97]
CC$_{UCLUST}$	PD Range	[164.79 – 217.72]	[169.41 – 198.36]	[109.33 – 109.33]	[2.37 – 47.13]	[0.52 – 1419.31]	[3.02 – 3.81]	[4.61 – 53.29]
	Shannon Range	[2.61 – 3.83]	[2.92 – 3.91]	[6.41 – 6.41]	[0.93 – 7.13]	[3.26 – 7.61]	[3.72 – 7.38]	[0.13 – 6.17]
	Simpson Range	[0.64 – 0.87]	[0.56 – 0.93]	[0.96 – 0.96]	[0.081 – 0.99]	[0.84 – 0.99]	[0.21 – 0.99]	[0.09 – 0.98]
SUMACLUST	PD Range	[106.00 – 162.78]	[142.85 – 174.19]	[89.22 – 89.22]	[0.93 – 39.47]	[0.59 – 1279.29]	[2.98 – 3.29]	[0.28 – 49.61]
	Shannon Range	[2.00 – 3.01]	[2.19 – 3.28]	[5.48 – 5.48]	[0.16 – 7.43]	[2.32 – 7.32]	[1.00 – 7.89]	[1.37 – 7.83]
	Simpson Range	[0.52 – 0.73]	[0.54 – 0.75]	[0.95 – 0.95]	[0.027 – 0.98]	[0.80 – 0.99]	[0.40 – 0.98]	[0.19 – 0.91]
CC$_{SUMACLUST}$	PD Range	[114.96 – 171.57]	[147.85 – 187.91]	[93.81 – 93.81]	[0.86 – 41.63]	[0.81 – 1292.34]	[1.37 – 4.89]	[0.18 – 51.35]
	Shannon Range	[2.51 – 3.94]	[2.96 – 4.11]	[5.94 – 5.94]	[1.21 – 7.13]	[3.12 – 7.79]	[2.17 – 7.25]	[1.91 – 7.39]
	Simpson Range	[0.68 – 0.79]	[0.51 – 0.74]	[0.96 – 0.96]	[0.06 – 0.99]	[0.88 – 0.99]	[0.23 – 0.98]	[0.27 – 0.88]
SWARM	PD Range	[18.37 – 24.73]	[17.36 – 19.81]	[30.84 – 30.84]	[1.44 – 28.66]	[0.54 – 706.57]	[5.79 – 6.18]	[2.06 – 39.71]
	Shannon Range	[2.98 – 3.91]	[2.01 – 3.04]	[5.03 – 5.03]	[0.28 – 7.63]	[1.0 – 7.79]	[1.66 – 7.81]	[1.47 – 6.91]
	Simpson Range	[0.70 – 0.82]	[0.53 – 0.74]	[0.95 – 0.95]	[0.05 – 0.98]	[0.50 – 0.99]	[0.00 – 0.99]	[0.18 – 0.89]
CC$_{SWARM}$	PD Range	[19.18 – 26.87]	[18.43 – 22.61]	[31.48 – 31.48]	[2.34 – 29.97]	[1.37 – 748.71]	[2.34 – 8.46]	[3.18 – 40.73]
	Shannon Range	[1.66 – 4.87]	[1.19 – 4.13]	[6.03 – 6.03]	[0.89 – 7.13]	[2.81 – 8.06]	[2.81 – 7.87]	[2.03 – 6.88]
	Simpson Range	[0.66 – 0.88]	[0.41 – 0.81]	[0.91 – 0.91]	[0.11 – 0.99]	[0.74 – 0.99]	[0.14 – 0.99]	[0.07 – 0.86]

Table shows ranges of values for Faiths Phylogeny Diversity (PD), Shannon and Simpson coefficient over all samples in a dataset in the format [*minimum* – *maximum*]. Most of these datasets contain multiple samples and Alpha diversity metrics like PD, Shannon and Simpson values are generated for each of these samples separately. Biodiversity metric values changes significantly over samples e.g. diversity from hair samples and teeth cavity are supposedly different. So instead of mean values this Table represents [*minimum* – *maximum*] ranges of values a sample can take. Similar ranges reflect similar diversity.

environmental shotgun reads. In *Annual International Conference on Research in Computational Molecular Biology*, pages 17–28. Springer, 2008.

[3] Nan et al. Alterations of the human gut microbiome in liver cirrhosis. *Nature*, 513(7516):59–64, 2014.

[4] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases*, VLDB '99, pages 518–529. Morgan Kaufmann Publishers Inc., 1999.

[5] Teeling Hanno and Glöckner Frank Oliver. Current opportunities and challenges in microbial metagenome analysisa bioinformatic perspective. *Briefings in bioinformatics*, 2012.

[6] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. pages 604–613, 1998.

[7] Andrew McCallum, Kamal Nigam, and Lyle H. Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 169–178. ACM, 2000.

[8] Tanaseichuk Olga, Borneman James, and Jiang Tao. Separating metagenomic short reads into genomes via clustering. *Algorithms for Molecular Biology*, 7(1):1, 2012.

[9] Mihai Pop and Steven L Salzberg. Bioinformatics challenges of new sequencing technology. *Trends in Genetics*, 24(3):142–149, 2008.

[10] Zeehasham Rasheed and Huzefa Rangwala. *MC-MinH: Metagenome Clustering using Minwise based Hashing*, chapter 74, pages 677–685.

[11] Zeehasham Rasheed, Huzefa Rangwala, and Daniel Barbara. Efficient clustering of metagenomic sequences using locality sensitive hashing. In *SDM*, pages 1023–1034. SIAM, 2012.

[12] Blaxter et al. Defining operational taxonomic units using dna barcode data. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1462):1935–1943, 2005.

[13] Bokulich et al. Quality-filtering vastly improves diversity estimates from illumina amplicon sequencing. *Nat Methods*, 10:57–59, 2013.

[14] C. Mercier et al. Sumatra and sumaclust: fast and exact comparison and clustering of sequences [submitted for publication]. 2014.

[15] Caporaso et al. Qiime allows analysis of high-throughput community sequencing data. *Nat Meth*, 7(5):335–336, 2010.

[16] Chakravorty et al. A detailed analysis of 16s ribosomal rna gene segments for the diagnosis of pathogenic bacteria. *Journal of microbiological methods*, 69(2):330–339, 2007.

[17] Costello et al. Bacterial community variation in human body habitats across space and time. *Science*, 326(5960):1694–1697, 2009.

[18] Daniel P Faith. Conservation evaluation and phylogenetic diversity. *Biological conservation*, 61(1):1–10, 1992.

[19] DeSantis et al. Greengenes, a chimera-checked 16s rrna gene database and workbench compatible with arb. *Applied and environmental microbiology*, 72(7):5069–5072, 2006.

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

[21] Kopylova et al. Open-source sequence clustering methods improve the state of the art. *mSystems*, 1(1), 2016.

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

[23] Mahé et al. Swarm v2: highly-scalable and high-resolution amplicon clustering. *PeerJ*, 3:e1420, 2015.

[24] Neufeld et al. Open resource metagenomics: a model for sharing metagenomic libraries. *Standards in Genomic Sciences*, 5:203–210, 2011.

[25] Ondov et al. Mash: fast genome and metagenome distance estimation using minhash. *bioRxiv*, 2016.

[26] Pruesse et al. Silva: a comprehensive online resource for quality checked and aligned ribosomal rna sequence data compatible with arb. *Nucleic acids research*, 35(21):7188–7196, 2007.

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

[28] Ramirez et al. Biogeographic patterns in below-ground diversity in new york city's central park are similar to those observed globally. *Proceedings of the Royal Society of London B: Biological Sciences*, 281(1795), 2014.

[29] Schloss et al. Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, 75(23):7537–7541, 2009.

[30] Turnbaugh et al. The human microbiome project: exploring the microbial part of ourselves in a changing world. *Nature*, 449(7164):804, 2007.

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

[32] Thomas SB Schmidt, João F Matias Rodrigues, and Christian Von Mering. Ecological consistency of ssu rrna-based operational taxonomic units at a global scale. *PLoS Comput Biol*, 10(4):e1003594, 2014.

[33] Qiong Wang, George M Garrity, James M Tiedje, and James R Cole. Naive bayesian classifier for rapid assignment of rrna sequences into the new bacterial taxonomy. *Applied and environmental microbiology*, 73(16):5261–5267, 2007.