# Clustering Metagenomic Sequences Using Canopies

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

***Abstract*—The advancement in genome technologies has allowed for the collective sequencing of co-existing microbial communities (Metagenomics), omnipresent across various environments like the soil, ocean and human body. Metagenomics has spurred the development of several bioinformatics approaches for analyzing the diversity, abundance, function and role of the different organisms within these communities. As an example, metagenomics allows for the analysis of the pathogenic role played by the microbial organisms within the human host.**

**We have developed a novel and efficient clustering algorithm for analyzing large-scale metagenomic data. Our approach achieves efficiency by first partitioning the large number of sequence reads into groups (called canopies) using a fast locality sensitive hashing based distance function. This initial approximate assignment of reads to canopies is refined by using state-of-the-art sequence clustering algorithms. This two-phase approach allows for the use of our developed algorithms as a pre-processing phase for computationally expensive clustering algorithms. Using the clusters as surrogates for Operational Taxonomic Units (OTUs) we provide an estimate of the biodiversity within a community sample.**

**The Canopy-based clustering algorithm is evaluated on synthetic and real world 16S metagenome benchmarks. We demonstrate the ability of our proposed approach to efficiently determine meaningful OTU assignments and report significant speedup with regards to run time when combined with three different state-of-the-art sequence clustering algorithms.**

**Source Code: XXX Link Here.**

***Keywords*-Clustering, Canopy, Metagenomes, 16S, Biodiversity**

## I. Introduction

A large portion of the earth's biomass comprises trillions of tiny microorganisms with varying biodiversity and play a crucial role in ensuring the stability of environment. Communities of interacting microbes exists in several ecosystems varying from the soil, ocean and human body [**?**]; and though, most of them are beneficial to the host some are known to cause unwanted conditions and can be linked to cause of diseases in the host [**?**].

*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 clinical and ecological environments [**?**][**?**].

State-of-the-art genome 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 (based on technology) in random order. Sequencing communities of genomes exacerbates the problem since reads from different microbes are mixed together and upfront the composition of a community in terms of abundance and identity of bacteria is unknown [**?**]. Further, sequencing technologies produce datasets that range from Gigabytes (GB) to Terabytes (TB) with their own idiosyncrasies. Alternatively, targeted metagenome sequencing that involves sequencing of *marker genes* has been popular for the characterization of these communities. 16S and 18S sequences are repetitive marker genes within microbial genomes but have variations in their sequences that allow them to be separated into different taxonomic groups [**?**].

Over the years, several unsupervised clustering algorithms have been developed and used for the analysis of large-scale targeted and whole metagenome sequences (reviewed in Section II). The unsupervised grouping of similar sequences from the 16S/18S genes is referred as *binning* and allows for the estimation of biodiversity within a sample. Binning leads to assignment of similar sequences within groups called *Operational Taxonomic Units (OTUs)* [**?**]. Microbial OTUs are generally ecologically consistent across the hosts regardless of OTU clustering approaches [**?**].

In this paper we develop an approximate but efficient clustering algorithm based on Canopy Clustering [**?**] which can be used as a pre-clustering step with any state-of-the-art sequence clustering algorithms. Specifically, our approach identifies canopies with a greedy procedure and a fast sequence distance metric based on locality sensitive hashing [**?**]. Sequences within the canopies are considered as an initial partition (or grouping) of data which can be further refined by applying expensive and accurate clustering algorithms. Our developed approach treats each canopies independently allowing for easy parallelization.

We present experimental results on real and synthetic metagenome sequence benchmarks and show that use of Canopy Clustering (CC) in combination with UCLUST [**?**], SWARM [**?**], SUMACLUST [**?**] and Minimum Spanning Tree (MST) based clustering leads to improved run time efficiency with accurate clustering results. We also show the performance of our developed approach is scalable for large datasets with increased number of processors. The source code is freely available on Github[1] for public use.

## II. Literature Review

Over the years several sequence clustering methods have been developed and used widely for metagenome sequences. A comprehensive survey by Kopylova et. al. [?] benchmarks various sequence clustering algorithms including UCLUST [?], SWARM [?], SUMACLUST [?] and MOTHUR [?]

CD-HIT [?] is general purpose sequence clustering algorithm that follows an incremental, greedy approach. CD-HIT uses pairwise sequence alignment to find similar sequences. UCLUST [?] 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 [?] utilizes an efficient locality sensitive based hashing function to approximate the pairwise sequence similarity. MC-MinH [?] uses min-wise [?] hashing along with the greedy clustering to group 16S and whole metagenome sequences. Mash [?], 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 [?], AbundanceBin [?] and CompostBin [?]. 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 [?] uses a pairwise distance matrix as input and performs hierarchical clustering. An expensive sequence alignment is used between all pairs of input sequences for computing distances.

UCLUST [?] and recently developed, SWARM [?] and SUMACLUST [?] are considered to be the state-of-the-art metagenome sequence clustering methods by a benchmarking study [?]. SWARM uses an exhaustive single-linkage clustering approach with a fast pairwise sequence alignment. SUMACLUST is similar to UCLUST. Abundance ordered list of input sequences are compared against the representative set of already chosen sequences in SUMACLUST.

In this study we developed an approximate but highly efficient sequence clustering algorithm using Canopies [?]. The developed approach can be used as a pre-processing step for other state-of-art sequence clustering algorithms to improve efficiency and also retains correctness in clustering.

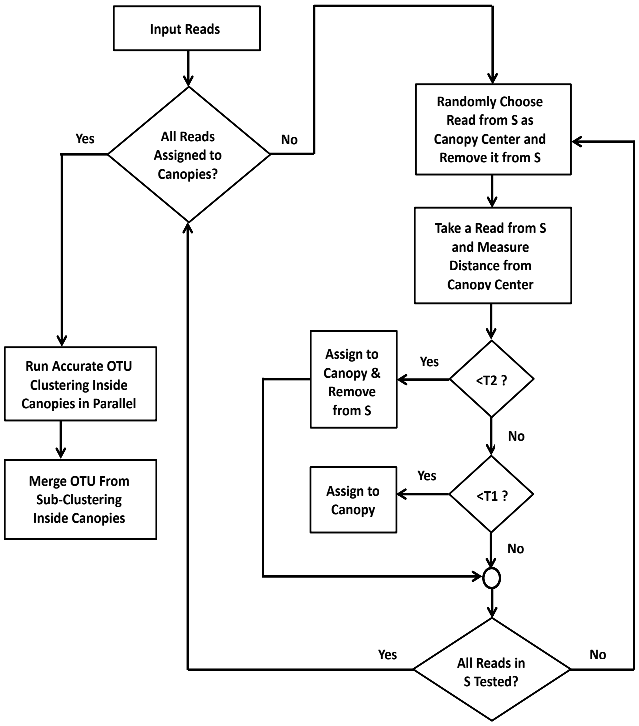

Input Reads
All Reads Assigned to Canopies?
Yes
No
Randomly Choose Read from S as Canopy Center and Remove it from S
Take a Read from S and Measure Distance from Canopy Center
Run Accurate OTU Clustering Inside Canopies in Parallel
Assign to Canopy & Remove from S
Yes
<T2 ?
No
Merge OTU From Sub-Clustering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

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

## III. Methods

### A. *Overview*

Figure 1 shows an overview of our proposed canopy clustering approach for metagenome sequences. Sequence reads are represented with *kmers* defined as contiguous subsequence of length k. We select a read randomly as the canopy centroid. Then we compute the distances of other reads to this canopy centroid. Two thresholds are used in Canopy clustering algorithm. If the a distance is less than a *tight* threshold ($T2$) we assign the read to the canopy and that read will not be considered for another canopy. Otherwise, if a distance is less than a *soft* threshold ($T1$) we assign the read to the canopy but that read will be available for another canopy. This process continues until all sequence reads is assigned to at least one canopy. For fast canopy assignment we use a random projection based technique. Specifically, we use Locality Sensitive Hash (LSH) to compute pairwise distances. More accurate clustering methods are used to sub-cluster each canopy in parallel. Sub-clustering method considers only the members within the canopy which reduces pairwise distance calculations. In this study UCLUST, SUMACLUST, SWARM and Minimum Spanning Tree (MST) based clustering are used as more accurate sub-clustering methods for canopies which are briefly discussed in Section III-D.

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

### B. *Canopy Clustering*

Canopy Clustering [?] is an efficient approximate clustering algorithm often used as pre-processing step for other accurate and expensive clustering methods like the K-means algorithm or the Hierarchical clustering algorithm. It is intended to speed up clustering operations on large data sets, where standard clustering algorithms may be impractical in terms of run time and memory consumption due to the large number of pairwise distance calculations. For a dataset with $N$ instances, worst case calculations without canopy clustering is $N^2$. After canopy clustering, if the number of canopies is $k$ then worst case calculations with canopy clustering is $\sum_{i=1}^{k}(c_i)^2$ where $c_i$ is the number of instances within $i$th canopy.

Canopy clustering uses two distance thresholds, (i) *soft* threshold $T1$ and (ii) *tight* threshold $T2$. If data point $p_1$ is within the soft distance threshold $T1$ with centroid $p_2$ then $p_1$ will reside in same canopy as $p_2$ but $p_1$ may belong to other canopies assuming that it has only met soft threshold and best match is yet to be found. Thus one data point may belong to multiple canopies in Canopy clustering. On the other hand if data point $p_1$ is within the tight distance threshold $T2$ with centroid $p_2$ then canopy clustering assigns $p_1$ to the same canopy as $p_2$ and stops assigning $p_1$ to any other canopy assuming that tight threshold has been met and best canopy assignment for $p_1$ has been found. Canopy centrists are selected randomly until all data points is assigned to at least one canopy.

### C. *Locality Sensitive Hashing*

Canopies are intended to reduce pairwise distance calculations. Canopy clustering itself should be efficient which requires a fast and approximate distance measure for Canopy assignments. Locality Sensitive Hashing (LSH) [?] is an algorithm for solving the approximate or exact Near Neighbor Search in high dimensional spaces. LSH reduces data dimensionality by randomly projecting data points to low dimensional bit signatures and provides approximate distance measure. These characteristics make LSH appropriate for dimensionality reduction and relatively accurate distance measure for Canopy clustering in metagenome data.

In this study we have chosen a random projection based hashing function that projects a $d$ dimensional data point into a $n$ dimensional bit representation where $n < d$ and reduces dimensionality. Given a random projection $v'$ of size $1 \times d$ and a data vector $a$ of size $1 \times d$, the dot product between them provides a scalar value. Sign of this scalar value represents which side of the random hyperplane does the data point exist. This information can be represented with a single bit. This way a single data point can be represented with limited number of bits. Hamming distance between these binary representations indicates *disagreements* between two data points.

### D. *Sub-Clustering Inside Canopies*

We have used three recent and popular sequence clustering methods as the expensive clustering measure inside canopies in this study. UCLUST [?], SUMACLUST [?] and SWARM [?] were used for sub-clustering canopies. We also implemented a Minimum Spanning Tree (MST) based sequence clustering method for accurate sub-clustering within canopies.

*1) UCLUST:* UCLUST follows a greedy process and creates *seeds* of sequences which generate clusters based on percent identity.

*2) SUMACLUST:* It follows similar approach as UCLUST. Based on greedy strategy SUMACLUST incrementally constructs clusters by comparing an abundance-ordered list of input sequences against the representative set of already-chosen sequences. Initially this list is empty.

*3) SWARM:* SWARM uses exhaustive single-linkage clustering based on optimal sequence alignment. Sequences that are less than a certain distance from any other other sequence in the cluster are clustered together. SWARM attempts to reduce the impact of clustering parameters on the resulting OTUs by avoiding arbitrary global clustering thresholds and input sequence ordering dependence. At first SWARM builds an initial set of OTUs is constructed by iteratively agglomerating similar amplicons. Then amplicon abundance values are used to reveal OTUs internal structures and to break them into sub-OTUs.

*4) Minimum Spanning Tree (MST):* Minimum Spanning Tree (MST) produces a connected spanning tree of undirected graph. It connects all vertices together such that the total weight of edges remains minimum. This property has been used for clustering [?] [?]. We represent edges with LSH based distances. Then edges with highest distances were removed from MST to generate OTUs.

### E. *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 OTU representations generated by 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. We also use Minimum Spanning Tree and Locality Sensitive Hashing (LSH) distance based clustering method to cluster OTU representations. A brief description of MST based clustering is given in Section III-D. Unlike merging with greedy clustering methods like UCLUST, this approach requires number of edges to remove as prior information.

In the following Sections UCLUST, SUMACLUST, SWARM and MST with Canopy clustering approach are represented as CC$_{UCLUST}$, CC$_{SUMACLUST}$, CC$_{SWARM}$ and CC$_{MST}$ respectively where the term CC stands for Canopy Clustering.

## IV. Experimental Evaluation

### A. *Dataset Description*

To evaluate the performance of our proposed approach we use previously published synthetic and real world sequence benchmarks [?] which are publicly available[2]. We used three synthetic 16S rRNA gene mock community datasets (Bokulich$_2$, Bokulich$_3$, and Bokulich$_6$) from Bokulich et. al. [?] and three real data sets: a 16S rRNA gene soil data set (Canadian Soil) [?], a 16S rRNA gene human data set (Body Sites) [?] and 18S rRNA gene soil data set (Global Soil) [?]. Key statistics and relevant information regarding these datasets are presented in Table I.

Table I
Dataset Statistics

| Datasets | Type | # of | # of | Read | Platform |
|---|---|---|---|---|---|
| | | Reads | Samples | Length | |
| Bokulich$_2$ [?] | M | 6,938,836 | 4 | 189–251 | H |
| Bokulich$_3$ [?] | M | 3,594,237 | 4 | 114–151 | H |
| Bokulich$_6$ [?] | M | 250,903 | 1 | 114–150 | H |
| Canadian Soil [?] | R | 2,966,053 | 13 | 76–10 | H |
| Body Sites [?] | R | 886,630 | 602 | 117–351 | G |
| Global Soil [?] | R | 9,252,764 | 57 | 119–151 | H |

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

**Synthetic Datasets:**

*1) Bokulich$_2$:* This dataset was prepared using the Illumina TruSeq v2 paired–end library preparation kit. It is a simulated 16S rRNA gene microbial community data set. This dataset contains 19 taxonomic Families, 19 Genera, 22 Species and 22 Strains in total. This dataset can also be found in the QIIME database (identifier 1685).

*2) Bokulich$_3$:* Similar to Bokulich$_2$ except that it was prepared with the TruSeq v1 paired-end library kit at Illumina Cambridge and is also available in the QIIME database (identifier 1686).

*3) Bokulich$_6$:* This 16S rRNA dataset was 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. All these datasets from Bokulich et al. [?] are available at QIIME database[3] under their respective ID's. Since, these are simulated datasets the taxonomic profile of microbial organisms within them are known.

**Real World Datasets:**

*4) Canadian Soil:* The Canadian Soil dataset[4] contains genomic data of soil spanning from Arctic Tundra to Agricultural soil suitable for different agricultural products.

*5) Body Sites:* This dataset contains composition of 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.

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

### B. *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.

*1) Faiths phylogenetic diversity metric (PD):* Faiths phylogenetic diversity [?] combines all the branch lengths of phylogenetic tree as a measure of diversity. So, 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.

*2) Shannon Entropy:* Shannon-Wiener diversity index is defined as:

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

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. The fact that the index incorporates both components of biodiversity can be seen as both a strength and a weakness. It is a strength because it provides a simple summary, but it is a weakness because it makes it difficult to compare communities that differ greatly in richness.

*3) Simpson's Index:* Simpsons index is defined as $1 - dominance$ or

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

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. It measures *evenness* of the community from 0 to 1. Higher value of this index implies higher similarity and relatively lower diversity of microorganisms within a sample.

*4) F-measure:* In case of synthetic datasets, false-positive (FP; taxonomy/OTU exists in observed but not expected), false-negative (FN; taxonomy/OTU exists in expected but not observed), and true-positive (TP; taxonomy/OTU exists in both observed and expected) measures were computed from cluster output and the ground truth which is the

[2] https://github.com/ekopylova/otu-clustering

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

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

Table II
PERFORMANCE COMPARISON [$F$-SCORE AND PEARSON CORRELATION COEFFICIENT ($\rho$)]

| Methods | Comparison Metric | Datasets | | | | | |
|---|---|---|---|---|---|---|---|
| | | *Synthetic* | | | *Real World* | | |
| | | Bokulich$_2$ | Bokulich$_3$ | Bokulich$_6$ | Body Sites | Canadian Soil | Global Soil |
| UCLUST | $F$-Measure | 0.39 | 0.40 | 0.51 | $N/A$ | $N/A$ | $N/A$ |
| CC$_{UCLUST}$ | $F$-Measure | ***0.39*** | ***0.41*** | ***0.52*** | $N/A$ | $N/A$ | $N/A$ |
| | ($\rho$) with Respect to UCLUST | ***0.9831*** | ***0.9753*** | ***0.9831*** | ***0.9682*** | ***0.8419*** | ***0.9824*** |
| SUMACLUST | $F$-Measure | 0.40 | 0.41 | 0.51 | $N/A$ | $N/A$ | $N/A$ |
| CC$_{SUMACLUST}$ | $F$-Measure | ***0.41*** | ***0.42*** | ***0.51*** | $N/A$ | $N/A$ | $N/A$ |
| | ($\rho$) with Respect to SUMACLUST | ***0.9709*** | ***0.9813*** | ***0.9538*** | ***0.9518*** | ***0.7643*** | ***0.8714*** |
| SWARM | $F$-Measure | 0.46 | 0.48 | 0.55 | $N/A$ | $N/A$ | $N/A$ |
| CC$_{SWARM}$ | $F$-Measure | ***0.46*** | ***0.49*** | ***0.56*** | $N/A$ | $N/A$ | $N/A$ |
| | ($\rho$) with Respect to SWARM | ***0.9817*** | ***0.97861*** | ***0.9251*** | ***0.9648*** | ***0.7581*** | ***0.9143*** |
| MST | $F$-Measure | 0.40 | 0.42 | 0.52 | $N/A$ | $N/A$ | $N/A$ |
| CC$_{MST}$ | $F$-Measure | 0.41 | 0.42 | 0.52 | $N/A$ | $N/A$ | $N/A$ |
| | ($\rho$) with respect to MST | 0.9301 | 0.9173 | 0.9081 | 0.8906 | 0.8132 | 0.8719 |
| | ($\rho$) with respect to UCLUST | 0.7461 | 0.8143 | 0.8272 | 0.7417 | 0.6935 | 0.7974 |
| | ($\rho$) with respect to SUMACLUST | 0.7182 | 0.6903 | 0.8613 | 0.8293 | 0.6791 | 0.8213 |
| | ($\rho$) with respect to SWARM | 0.7914 | 0.7352 | 0.8904 | 0.7619 | 0.7372 | 0.8502 |

Table shows values of $F$-measures and Pearson Correlation Coefficient ($\rho$-value) of UCLUST, SUMACLUST, SWARM, MST and their respective versions with Canopy clustering. $F$-measures is only available for synthetic datasets but not for real world datasets since no ground truths like known taxonomic profiles are available for them. $\rho$-value was calculated based on the taxonomy profiles at Genus level generated from clustered OTUs provided by a method and it's corresponding Canopy counterpart. Higher $F$-measures reflect better clustering by adhering to ground truth. Higher $\rho$-values reflect stronger correlation between taxonomic profiles. Higher F-scores and $\rho$ values are represented with bold and italic letters.

expected taxonomic composition. The following definitions were used:

$$precision = \frac{TP}{(TP + FP)} \tag{3}$$

$$recall = \frac{TP}{(TP + FN)} \tag{4}$$

$$FScore = \frac{2 \times precision \times recall}{(precision + recall)} \tag{5}$$

*5) Pearson Coefficient Correlation ($\rho$-value):* After getting Operational Taxonomic Units (OTU) 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.

### C. Parameter Settings

For kmers, the value of parameter k was set to 4. For Locality Sensitive Hashing the number of Hyper-planes (parameter $d$) was set to 150. For performance analysis and comparison we have chosen $T1 = 0.45$ and $T2 = 0.34$. We performed all the experiments on computers with Intel 5th generation Core i7 2.70GHz 64bit processor with 8 core CPUs and 12GB memory. Similarity identity of subclustering methods were set to default 97%. For implementation we used Python 2.7.12 and QIIME [?] version 1.9.0. Taxonomy for reported OTUs was assigned using the RDP Classifier [?] against the 97% representative databases for Greengenes [?] and Silva [?] for methods used in this study. We used PyNast[5] open source sequence aligner for aligning clustered output.

## V. RESULTS DISCUSSION

### A. Clustering Performance

Table II shows the performance of UCLUST, SUMACLUST and SWARM with their corresponding versions with CC clustering. Table II compares F-scores and Pearson Correlation Coefficient ($\rho$). F-scores are only available for synthetic benchmarks since taxonomic profile for them is known as ground truth. Correlation values were generated based on taxonomic profiles at Genus level generated from outputs from clustering methods. From Table II we can see that F-scores obtained from a clustering method and its corresponding Canopy clustering version are very similar and in some cases better. We see a higher F-score for Bokulich$_2$ benchmark from CC$_{SUMACLUST}$. CC$_{UCLUST}$ and CC$_{SWARM}$ provided same F-scores as their respective naive versions for Bokulich$_2$. For Bokulich$_3$ benchmark we observed higher F-scores for all Canopy clustering methods. Finally for Bokulich$_6$ benchmark F-scores of CC$_{UCLUST}$ and CC$_{SWARM}$ were improved comparing to respective naive versions.

For all benchmarks we observe strong correlations between taxonomic profiles at genus level. The highest correlation was observed for Bokulich$_6$ benchmark between UCLUST and CC$_{UCLUST}$ with 0.9831. CC$_{MST}$ provided

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

Table III
**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 |
| 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] |
| | 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] |
| | 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] |
| 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] |
| | 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] |
| | 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] |
| 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] |
| | 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] |
| | 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] |
| 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] |
| | 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] |
| | 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] |
| 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] |
| | 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] |
| | 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] |
| 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] |
| | 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] |
| | 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] |
| MST | PD Range | [9.14 − 91.06] | [11.43 − 92.27] | [60.70 − 60.70] | [1.18 − 33.61] | [1.92 − 826.64] | [2.35 − 5.81] |
| | Shannon Range | [0.76 − 4.08] | [1.64 − 2.98] | [4.72 − 4.72] | [1.96 − 6.11] | [2.14 − 7.29] | [2.38 − 6.07] |
| | Simpson Range | [0.12 − 0.71] | [0.21 − 0.69] | [0.90 − 0.90] | [0.11 − 0.92] | [0.49 − 0.94] | [0.11 − 0.98] |
| CC$_{MST}$ | PD Range | [10.43 − 89.61] | [12.83 − 92.41] | [63.21 − 63.21] | [1.46 − 34.15] | [2.73 − 824.62] | [1.81 − 5.37] |
| | Shannon Range | [0.96 − 3.91] | [1.21 − 3.16] | [4.91 − 4.91] | [2.02 − 6.51] | [2.72 − 7.83] | [2.17 − 6.29] |
| | Simpson Range | [0.14 − 0.68] | [0.32 − 0.73] | [0.92 − 0.92] | [0.07 − 0.94] | [0.58 − 0.94] | [0.14 − 0.98] |

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.

Table IV
**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 | MST | CC$_{MST}$ | Speed Up |
| *Synthetic* | Bokulich$_2$ | 6,938,836 | 12.71 | ***6.03*** | ***2.10x*** | 87.53 | ***33.03*** | ***2.65x*** | 128.12 | ***37.87*** | ***3.38x*** | 564.60 | ***294.36*** | ***1.91x*** |
| | Bokulich$_3$ | 3,594,237 | 8.91 | ***4.89*** | ***1.82x*** | 11.73 | ***7.89*** | ***1.48x*** | 9.10 | ***6.93*** | ***1.31x*** | 369.7 | ***204.61*** | ***1.80x*** |
| | Bokulich$_6$ | 250,903 | 1.21 | ***1.08*** | ***1.12x*** | ***1.28*** | 1.34 | 0.96 | 1.97 | ***1.29*** | ***1.53x*** | 252.81 | ***146.41*** | ***1.73x*** |
| *Real World* | Body Sites | 886,630 | 2.12 | ***1.51*** | ***1.40x*** | 15.42 | ***7.88*** | ***1.95x*** | 3.64 | ***1.71*** | ***2.13x*** | 307.2 | ***174.27*** | ***1.76x*** |
| | Canadian Soil | 2,966,053 | 9.55 | ***5.91*** | ***1.62x*** | 363.96 | ***81.18*** | ***4.48x*** | 97.50 | ***32.16*** | ***3.03x*** | 472.43 | ***254.91*** | ***1.85x*** |
| | Global Soil | 9,252,764 | 72.47 | ***43.53*** | ***1.67x*** | 510.92 | ***66.27*** | ***7.70x*** | 269.07 | ***56.14*** | ***4.79x*** | 988.93 | ***464.29*** | ***2.13x*** |

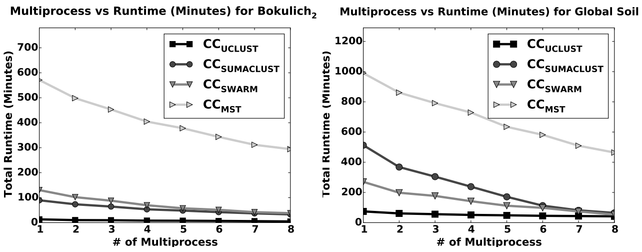
relatively similar F-scores and high $\rho$ values indicating positive correlations of taxonomic outcomes with other methods. From observed F-scores and correlation values we can say that running a sequence clustering method with our proposed LSH based Canopy cluster framework can bring similar or better results.

### *B. Biodiversity Estimation*

Clustering metagenome sequences outputs OTUs that represent biodiversity contained in the samples from which data is collected. We compare the biodiversity represented by OTUs from naive methods and their respective Canopy clustering variations used in this study. Table III shows Faiths phylogenetic diversity metric (PD), Shannon and Simpson index after clustering with different methods. These metrics are some of the popular Alpha Diversity metrics meaning that they measure species diversity in sites or habitats at a local scale. These metric values are generated per sample basis. Table III shows ranges of metric values in $[minimum - maximum]$ format provided by methods with and without Canopy clustering. Any sample of a dataset will take value from this range. These ranges may not be same since OTUs vary over clustering methods. But they should be similar for a method and it's Canopy counterpart. We can see from Table III that Canopy clustering based methods produce similar ranges of values as their naive counterparts. No significant changes in diversity metric values were observed which indicates that Canopy based approaches reproduce similar biodiversity while reducing run time.

### *C. Runtime Comparison*

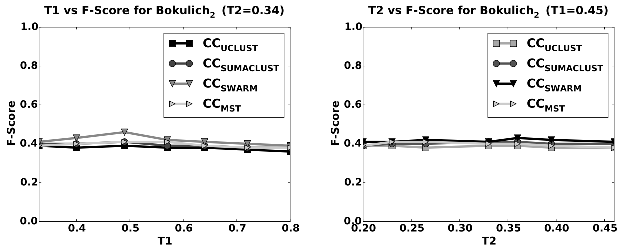
Table IV shows the run time in minutes of UCLUST, SUMACLUST, SWARM, MST and their respective ver-



(a)

(b)

Figure 2. Effect of number of multiprocessor on Runtime of CC$_{UCLUST}$, CC$_{SUMACLUST}$, CC$_{SWARM}$ and CC$_{MST}$ for Bokulich$_2$ and GLobal Soil datasets, two of the largest datasets used in this study.



(a)

(b)

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

sions with our proposed Canopy clustering pipeline. We see from Table IV that CC$_{UCLUST}$ outperforms UCLUST for all datasets. The highest speed up for UCLUST was observed for Bokulich$_2$ dataset where CC$_{UCLUST}$ was 2.10 times faster than UCLUST. CC$_{SUMACLUST}$ outperforms SUMACLUST in most cases. Major improvement in run time was observed in large scale real world metagenome datasets specially Canadian Soil and Global Soil datasets. The highest speed-up for CC$_{SUMACLUST}$ was observed to be 7.70x for Global Soil dataset. We also see speedups for CC$_{SWARM}$ and CC$_{MST}$ from Table IV. The highest speedups for CC$_{SWARM}$ and CC$_{MST}$ were 4.79x and 2.13x for Global Soil dataset which is the largest benchmark used in this study, respectively. From these observations we conclude that Canopy clustering not only reduces the run time of an expensive clustering method but also scales well for larger benchmark datasets comparing to the smaller ones.

### D. *Effect of Varying Number of Multi-processes*

Figure 2a-2b shows the runtime of CC$_{UCLUST}$, CC$_{SUMACLUST}$, CC$_{SWARM}$ and CC$_{MST}$ on two largest benchmarks titled Bokulich$_2$ and Global Soil. We observe that increasing the number of multiprocessors reduces the total runtime. Steeper curves show reduction in runtime with increasing number of multiprocessors. Significant reductions in run time were observed CC$_{MST}$, CC$_{SUMACLUST}$ and CC$_{SUMACLUST}$ compared to CC$_{UCLUST}$.

### E. *Sensitivity Analysis (Varying T1 and T2)*

Figure 3a-3b shows effect of varying $T1$ ($T2$ fixed at 0.34) and $T2$ ($T1$ fixed at 0.45) on F-scores for the largest synthetic dataset Bokulich$_2$. 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 can say 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 can say that when $T2$'s range is in 0.25 to 0.35 our proposed approach achieves better F-Scores. Lower $T2$ may lead to higher runtime since instances will continue to reappear until the $T2$ is met.

## VI. Conclusion and Future Work

We developed a greedy approximate clustering process for any accurate and relatively expensive clustering on large scale metagenome datasets. Our approach takes advantage of the multi-core CPU systems by partitioning the large dataset with a fast and cheap pairwise distance measure and then deploying comparatively expensive clustering in parallel which considers only data points that are inside a partition. Our proposed approach scales well with large datasets and provide significant reduction in computation time. We demonstrate that our approach provides similar outcome in terms of biodiversity metrics, ground truth and taxonomic correlation with corresponding expensive clustering methods.