LSH-Div: Species Diversity Estimation using Locality Sensitive Hashing

Zeehasham Rasheed
Department of Computer Science
George Mason University
Fairfax, VA 22030 USA
zrasheed@gmu.edu

Huzefa Rangwala
Department of Computer Science
George Mason University
Fairfax, VA 22030 USA
rangwala@cs.gmu.edu

Daniel Barbará
Department of Computer Science
George Mason University
Fairfax, VA 22030 USA
dbarbara@gmu.edu

***Abstract*—Metagenome sequencing projects attempt to determine the collective DNA of organisms, co-existing as communities across different environments. Computational approaches analyze the large volumes of sequence data obtained from these ecological samples, to provide an understanding of the species diversity, content and abundance. In this work we present a scalable, species diversity estimation algorithm that achieves computational efficiency by use of a locality sensitive hashing algorithm (LSH). Using fixed-length, gapless subsequences, we improve the sensitivity of pairwise sequence comparisons. Using the LSH-based function, we first group similar sequences into bins commonly referred to as operational taxonomic units (OTUs) and then compute several species diversity/richness metrics. The performance of our algorithm is evaluated on synthetic data and eight targeted metagenome samples obtained from the seawater. We compare our results to three state-of-the-art diversity estimation algorithms. We demonstrate the strength of our approach in terms of computational runtime and effective OTU assignments.**

The source code for LSH-Div is available at the supplementary website under the GNU GPL license. Supplementary material is available at http://www.cs.gmu.edu/~mlbio/LSH-DIV

***Keywords*-16S metagenomics, species diversity, clustering**

I. Introduction

Latest sequencing technologies have allowed researchers to study the "biodiversity" of microbial organisms, existing as communities across various environmental and clinical samples [1]. Several computational approaches have been developed to analyze the sequence data obtained from the targeted metagenomic (16S marker genes) or "whole" metagenomic studies [2]. Given these ecological samples, researchers estimate the species diversity, content and abundance using computational approaches.

Two class of approaches have been widely used for estimating species diversity from environmental sequence samples. The first set of approaches classify sequences into taxonomic classes using a reference database [3], [4], [5], and are referred to as "phylogenetic" or "taxonomic" classification methods. The second set of approaches assign sequences with shared similarity to a common group/bin. These bins are referred to as operational taxonomic units (OTUs) [6], [7], [8].

Using the OTU assignments allows for estimation of species richness measures like Chao1 index, Shannon diversity index and Abundance-based Coverage Estimator (ACE) index. Mothur, DOTUR and ESPRIT are examples of commonly used algorithms for OTU estimation [9], [10].

We present a scalable OTU Estimation algorithm called LSH-Div for targeted metagenomic sequences (or called 16S metagenomes). The central idea of our approach is the use of an efficient randomized search technique called "Locality Sensitive Hashing" (LSH) [11]. The LSH algorithm computes similarity based on randomly chosen sequence positions, that essentially compresses the length of input sequences. Buhler et. al. [12] use a similar LSH-based concept to determine all pairs of similar segments between two long genomes. We previously developed a metagenome clustering approach called MC-LSH [13]. We extend MC-LSH by using fixed-length gapless subsequences, referred to as w-mers to improve the sensitivity of matching pairs of sequences. Using w-mers per chosen position, helps in the identification of conserved regions and improves the accuracy of comparing sequence pairs. The LSH-Div algorithm groups sequences using the LSH function within a greedy, iterative clustering framework. After assigning sequences within a sample to different OTUs (or clusters), LSH-Div reports the standard species richness metrics.

We evaluate the performance of LSH-Div on eight different environmental samples obtained from the sea water, and known to have varying complexities of microbial diversity [14]. Our evaluation focuses on the quality of OTUs, richness estimators, computational run-time and pairwise sequence similarity of sequences within each OTU. We compare our LSH-Div algorithm to three state-of-the-art OTU estimation algorithms.

II. Related Work

1) DOTUR:: DOTUR [7] (Defining Operational Taxonomic Units and Estimating Species Richness) uses a pairwise distance matrix as input. The pairwise distances are computed after performing global alignment between all pairs of sequences in the input set. DOTUR takes the pairwise matrix as input, and uses a hierarchical clustering algorithm to determine relationships between input sequences

978-1-4673-2560-8/12/$31.00 ©2012 IEEE

(represented as a tree or dendogram). Three possible choices are available for merging clusters at each iteration within the hierarchical clustering frameworks: (i) nearest neighbor, (ii) furthest neighbor and (iii) unweighted pair group method with arithmetic mean (UPGMA).

An input parameter referred to as distance cutoff is used to split the dendogram at a specific level to produce different OTUs, and as such assignment of sequences to the OTUs. DOTUR reports the standard species richness metrics i.e., Shannon's, Chao1 and ACE index as a function of the distance cutoff. The distance cutoff parameter d, sets an upper bound, such that sequences within the a given OTU are at most d% distance apart.

2) Mothur:: Mothur [6] is very similar to DOTUR, and starts by calculating pairwise global alignment distance matrix for a given set of sequences. These sequences are then clustered to assign OTUs. In fact, Mothur uses the DOTUR implementation of clustering methods such as nearest neighbor, furthest neighbor and average neighbor. For varying distance cutoffs (d), Mothur reports the key statistics such as number of OTUs, number of sequences in the largest OTU, number of OTUs with only one sequence (singleton) and OTUs with two sequences (doubletons). Mothur also computes the standard species diversity metrics. Both, Mothur and DOTUR require the entire pairwise matrix to be loaded in memory.

3) ESPRIT:: Instead of using global alignment distances, ESPRIT [8] computes k-mer distance for each pair of input sequences to rapidly calculate pairwise distances. For each sequence, a complete genomic alphabet profile of selected k-mer is constructed and pairwise distance is derived based on that representation. To reduce the computational complexity, ESPRIT uses several strategies. If two sequences are identical or one sequence is a subset of the other, only the longer sequence is retained, and the number of occurrence of each retained sequence is recorded. This results in a reduced number of sequences for computing the k-mer pairwise distance matrix. After distance matrix is computed, ESPRIT performs furthest neighbor hierarchical clustering approach to define sequences into OTUs for different distance levels.

III. Methods

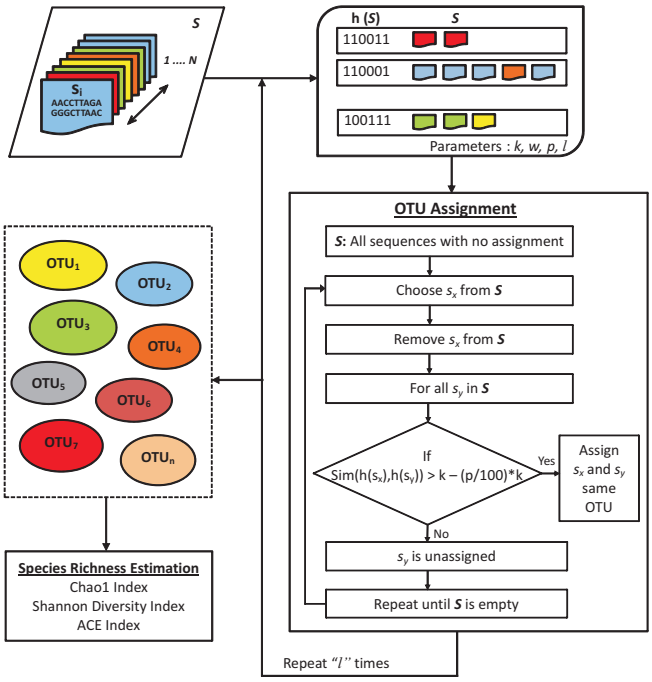
A. Locality Sensitive Hashing

Assume a set of N sequences, $\mathcal{S}$ obtained from a metagenome sample. Let s be the sequence of length n for which we construct a randomized hash function. We choose k uniform, random indices $i_1, \ldots, i_k$ in the range $\{1, \ldots, n\}$ to define a hash function $h(s)$ given by:

$$h(s) = \langle s[i_1], s[i_2], \ldots, s[i_k] \rangle \quad (1)$$

The function $h(s)$ in Equation 1 generates a concatenated string of length k (the number of chosen random indices) from the original string s of length n. Since DNA sequences

Figure 1: Process flow diagram of LSH-Div OTU estimation algorithm.



are made up of four nucleotides, the function $h(s)$ creates a mapping between the original 4^n dimensional space to a reduced 4^k dimensional space. The function $h(s)$ is called "locality-sensitive" [11] because the probability of two strings s_x and s_y mapped to the same hash value varies directly with their similarity. If the strings s_x and s_y are similar i.e., assumed to be differing by at the most p nucleotides, then the probability that they produce the same hash values is given by:

$$P[h(s_x) = h(s_y)] \geq \left(1 - \frac{p}{n}\right)^k. \quad (2)$$

Parameter p is the allowable mismatch factor between the two strings and $P[]$ is the probability which is computed over random choices of the k indices, $i_1, \ldots, i_k$. The LSH function also captures the partial global ordering of nucleotides that are present in the DNA sequences.

Using the LSH approach can lead to false positives and false negatives [13]. A false positive occurs when strings, s_x and s_y are mapped to same hash values even if they are different i.e., $h(s_x) = h(s_y)$. This happens due to the sampling of only those k nucleotide positions that are similar in two sequences. A false negative occurs when sequences s_x and s_y are similar but not 100% identical and $h()$ samples one or more of the k indices where s_x and s_y differ. This causes $h(s_x) \neq h(s_y)$ even if two sequences are very much similar. False negatives cannot be detected easily, but can be reduced substantially by multiple iterations of the sampling process. This is done by repeatedly choosing new sets of k

indices. The number of iterations for sampling different k indices is denoted by l.

Instead of using a single nucleotide (character) at the randomly chosen index i, we use a gapless subsequence of fixed length per index for our hash function. We refer to the subsequence of length w as a w-mer. The use of w-mer makes the hashing function more accurate and reduces false positives, because now we compare w-mers at the k different indices.

B. Algorithm

Figure 1 shows the flow chart for the LSH-Div algorithm. The algorithm has the following parameters: (i) k, the number of sampled indices for the hash key, (ii) l, the number of iterations of hash functions, (iii) w, the length of subsequence and p, the mismatch factor. These parameters influence the results in terms of number of OTUs, values of different richness estimators and the running time of our algorithm. The LSH-Div algorithm starts by using the LSH function $h(s)$ for the N input sequences in set $\mathcal{S}$. This function h() uses the k indices and the w-mers at the chosen k positions to determine the hash value (represented as a binary signature of length k) for each sequence. Once all the hash values are calculated, the procedure of OTU assignment is initiated.

The algorithm first chooses a random sequence and assigns the first OTU to that sequence. The algorithm uses the hamming distance denoted by $Sim(,)$, to compute the difference between the hash values, $h(s_x)$ and $h(s_y)$ of strings s_x and s_y, respectively. Using $Sim(,)$, we accept strings that differ by only a few substitutions and reject pairs that differ by many substitutions. Identical strings are always be accepted by this similarity function. We identify all unassigned sequences in $\mathcal{S}$ (that have no OTU assignments yet) which differ from the chosen sequence by at most p percentage of k. These sequences are then assigned to the same OTU and are removed from $\mathcal{S}$. We iterate through the same procedure until OTUs are assigned to all the sequences in $\mathcal{S}$. In case of w-mers, we expect exact match of the gapless fixed length subsequences per sampled position.

The LSH-Div Algorithm is repeated l times to assign same OTUs to similar sequences and improve the performance. For every iteration, a new LSH function chooses a different set of k indices. We proceed by performing a union operation on the OTU assignments obtained for the current iteration with the previous iteration. Our implementation is optimized, so as to merge OTUs from the previous iteration based on the similarity measured using the new set of k indices. For example, if sequences s_x and s_y were assigned to two different OTUs in the previous iteration and found to be similar in the current iteration, then the OTUs would be merged (unionized).

The most significant limitation with Mothur, DOTUR and ESPRIT algorithms is that they are computationally intensive. Unlike these methods which require pairwise distances for OTU assignments, LSH-Div algorithm does not involve any all-pairwise distances calculation for OTU estimation. This makes it faster and scalable in comparison to other approaches.

Table I: Environmental DNA samples used for OTU Estimation.

Sample	Site	Lat °N, Long °W	Depth	Temp	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. Lat is the latitude, Long is the longitude, Depth is in meters, Temp is in °C and Reads are the number of sequences in a sample.

The LSH-Div algorithm can handle sequences of varying lengths as well.To handle sequences of unequal lengths, we first identify the sequence with the shortest length, n'. Given, a sequence in the input set of length n we compute the LSH values for all $n - n'$ subsequences, each of the same length n'. To compare a pair of strings, we use the hamming distance between all pairs of subsequences of length n' generated from the two strings. Two strings are assumed to hash to the same values and belong to the same OTU, if any pairs of n' length subsequences satisfy the hamming-distance based similarity.

C. Computation of Pairwise Distance Cut-Offs

The distance cutoff parameter is used for diversity estimation algorithms. This parameter sets a bound on pairwise distances between the sequences within the OTUs. A distance cutoff of 0.03 indicates that all sequences within an OTU are at most 3% distant from all other sequences within that OTU. The OTU estimation in ESPRIT, Mothur and DOTUR are based on pairwise distances among sequence reads. The standard approach is to use the Needleman-Wunsch global alignment algorithm to optimally align each pair of sequences in a sample and compute pairwise distances. This operation is computationally expensive. ESPRIT computes the k-mer distance of each pair of sequences to rapidly calculate this distance.

The LSH-Div algorithm does not compute pairwise distances for OTU estimation. Therefore, in order to evaluate the performance of LSH-Div, we first determine the OTU assignments, and the compute pairwise distances between sequences within each OTU using the global alignment distance (or k-mer distance). We report the maximum pairwise

distance obtained, which ensures that all sequences within the OTUs are at most d% distant from each other.

IV. Materials and Implementation

A. Dataset Description

We evaluate the LSH-Div algorithm on synthetic and real environmental 16S metagenome samples. The synthetic data contains 345,000 short read sequences, generated by pyrosequencing of two PCR amplicon libraries from 43 known 16S rRNA gene fragments using the Roche GS20 system. This synthetic data is originally used in [15].

Environmental samples contain eight seawater samples taken from Sogin et. al. [14]. These samples use the 454 Life Sciences technology [16] to increase the number of sampled PCR amplicons in environmental surveys by orders of magnitude. The description of the samples are given in Table I. These sequences are of unequal length and the average sequence length for these datasets is 60 bp.

B. Species Richness Estimation Metrics

We compute three widely used statistical species richness estimators: (i) Chao1 index [17], (ii) Shannon Diversity index [18] and (iii) Abundance-based Coverage Estimator(ACE) [19](formulae not shown here). A brief explanation and mathematical expressions for these estimates are given in [6] and [7].

C. Hardware and Software Details

The LSH-Div software is implemented in Python. LSH-Div will be available from the supplementary website and will be provided with a GNU GPL license. All experiments were performed on a single workstation, with Intel-i5 2.53 GHz processor and 6 GB memory. Comparative approaches including Mothur, DOTUR and ESPRIT were also run on the same machine using the binary files made available by the authors of these programs.

V. Experimental Results

A. Synthetic Dataset

In order to show the correctness of LSH-Div algorithm, we evaluate the the algorithm on synthetic data. The goal of this experiment is to estimate the numbers of OTUs. Since this data is simulated from 43 reference gene sequences, we consider 43 number of OTUs as ground truth. Figure 2 shows the number of OTUs at various distance cutoff levels, also known as lineage-through-time curve in [7]. These results serve as ground truth, to benchmark the performance of different algorithms. To study how these algorithms perform in the presence of sequencing errors, we divide the synthetic dataset into two sets by keeping only those reads that have less than 3% or 5% errors. From Figure 2 we observe that all algorithms overestimate the number of OTUs for some distance cutoff levels. We can see that LSH-Div efficiently converges to the true number of OTUs (ground truth) at 0.04 cutoff for the reads with less than 3% error. This shows the correctness and effectiveness of LSH-Div algorithm even in presence of sequencing errors.

B. Environmental Samples

Table II shows the performance of LSH-Div algorithm on eight environmental samples at different global alignment distance cutoffs. The large number of OTUs with higher Chao1, Shannon diversity and ACE indexes indicate the presence of an environmental sample with large number of species and rich diversity. The complete results for each global-alignment distance cutoff and different LSH-Div parameter settings are within the supplementary paper. Figures 3a, 3b and 3c show the performance of different diversity estimation algorithms on number of OTUs, Chao1 index and ACE index for sample 53R, respectively.

C. Computational Complexity

Table III shows the run times for the different methods on each of the eight samples. Samples, FS312 and FS396 were trimmed because the original samples required more than 15 GB memory to compute and load the pairwise distance for Mothur and DOTUR. The trimming is done by ESPRIT filtering process. We report the run time results for the trimmed and original set of sequences. In Table III we also report the average computational time across eight samples for each method. We can see that on an average the LSH-Div algorithm is 2 times faster than ESPRIT, 18 times faster than DOTUR and 32 times faster than Mothur.

Table III: Runtime comparison.

Sample	# Reads	ESPRIT (sec)	Mothur (sec)	DOTUR (sec)	LSH-Div (sec)
53R	11218	283	10130	5129	**161**
55R	8680	266	5940	3511	**183**
112R	11132	537	12303	5567	**317**
115R	13441	348	13501	9237	**188**
137	12259	280	12861	6563	**172**
138	11554	296	12310	5618	**175**
FS312(t)	4002	205	2224	1990	**112**
FS396(t)	3457	192	1583	1457	**101**
Avg. Time	(8 samples)	300.87	8856.50	4884.00	**176.12**
FS312	52569	2980	–	–	**1436**
FS396	73657	3326	–	–	**1660**

Running time of LSH-Div, ESPRIT, Mothur and DOTUR performed on eight different samples. FS312(t) is trimmed to 4002 sequences and FS396(t) is trimmed to 3467 sequences. Avg. Time is the average computational time calculated across eight samples for each method. Mothur and DOTUR are unable to run FS312 and FS319 samples because of large number of sequences and memory requirements. LSH-Div parameter setting for this experiment is k = 30, w-mer = 3.

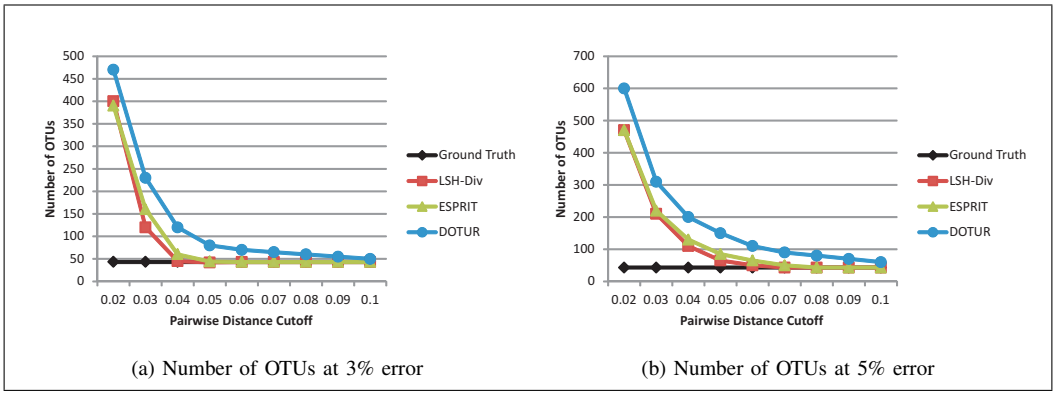
D. Evaluation of ESPRIT on Global Alignment Pairwise Distance.

In order to investigate why ESPRIT underestimates the number of OTUs and other species richness estimators, we

Table II: OTU and Species Richness Estimation by LSH-Div.

		OTUs Cutoff in Distance Units											
		0.03				0.05				0.10			
Sample	Reads	# OTUs	Chao1	H'	ACE	# OTUs	Chao1	H'	ACE	# OTUs	Chao1	H'	ACE
53R	11218	1459	3726.53	4.59	3564.58	1172	3050.32	4.22	2786.32	914	2308.14	3.95	2154.07
55R	8680	1461	3915.25	4.92	4274.4	1199	3296.38	4.53	3531.21	963	2418.55	4.35	2538.92
112R	11132	2111	6838.9	5.62	7255.16	1795	5781.85	5.19	6126.08	1506	4787.73	4.96	5040.54
115R	13441	1540	3930.08	4.6	3972.27	1205	3042.52	4.25	3094.09	943	2446.11	3.83	2409.63
137	12259	1266	3181.92	4.85	2740.44	1041	2595.72	4.60	2317.95	817	2028.76	4.24	1831.79
138	11554	1306	3031.06	4.61	2998.51	1072	2351.90	4.28	2372.84	845	1985.47	4.02	1827.27
FS312	52569	4321	13942.28	4.78	14345.15	3505	10367.35	4.56	10353.72	2771	7038.76	4.22	7444.19
FS396	73657	4594	14228.93	4.18	14826.74	3676	10672.02	4.04	10579.55	2876	7534.98	3.84	7370.95

Measurement of different diversity metrics estimated by LSH-Div algorithm for the samples characterized by Sogin et. al. [14]. The three OTU definitions 0.03, 0.05, and 0.10 are the cutoffs in distance units. OTU signifies the number of OTUs observed, Chao1 signifies the Chao1 estimate, H´ signifies the Shannon diversity index and ACE signifies the Abundance-based Coverage Estimator. LSH-Div parameter setting for this experiment is k = 30, w-mer = 3.



(a) Number of OTUs at 3% error

(b) Number of OTUs at 5% error

Figure 2: Number of OTUs for Synthetic Dataset. Lineage-through-time curves produced by LSH-Div, ESPRIT and DOTUR on synthetic data with each read containing up to (a) 3% and (b) 5% sequencing errors. Time taken by LSH-Div to produce these results is 578.62 secs where as ESPRIT takes 2892.1 secs to produce the same results. DOTUR results are taken from [8] because of computational issues in calculating pairwise distances of large number of sequences. LSH-Div parameter setting for this experiment is k = 30, w-mer = 3.

designed an experiment to evaluate LSH-Div and ESPRIT on different pairwise distance functions. We used ESPRIT's k-mer distance cutoffs to estimate the OTUs for different inputs varying from 0.01 to 0.10. As done for evaluation of LSH-Div, we first compute for sequences within each OTU, the maximum pairwise global alignment distance and then report the maximum of the scores across all the OTUs as the final result. In case of LSH-Div (no input distance cutoff parameter), we compute the maximum pairwise k-mer distance and the global alignment distances. These results are shown in Table IV. ESPRIT does not meet the distance cutoff levels when evaluated on global alignment distance. On the other hand, LSH-Div consistently meets the distance cutoff level on both global alignment and k-mer distances. The k-mer distance does not consider the secondary structure of sequences and produces inconsistent results. This was also discussed in related literature by Schloss [20].

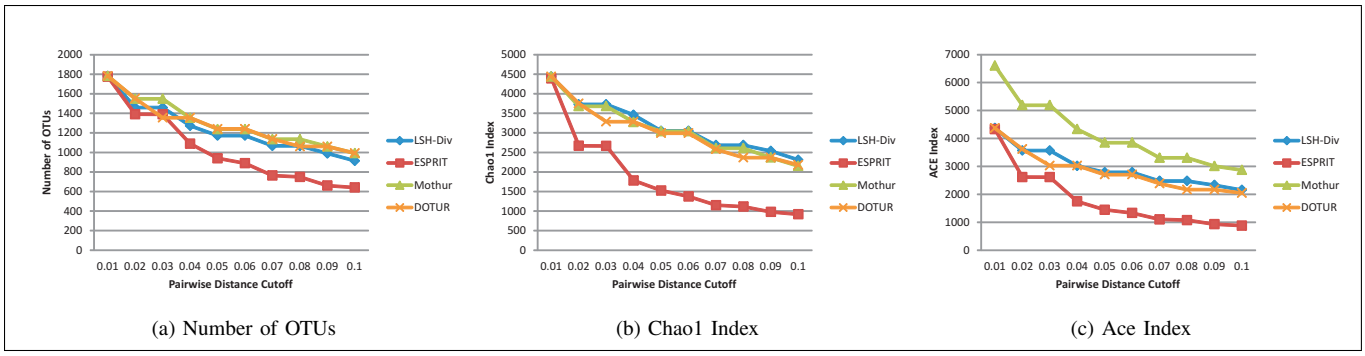
Table IV: ESPRIT and LSH-Div results for different cut-offs (Sample 53R)

	ESPRIT		LSH-Div	
k-mer distance	# OTUs	NW	# OTUs	NW / k-mer
0.01	1774	0.333	1782	0.0
0.02	1391	0.333	1459	0.016
0.03	1390	0.347	1459	0.016
0.04	1090	0.508	1273	0.033
0.05	940	0.546	1172	0.050
0.06	889	0.546	1172	0.050
0.07	764	0.546	1067	0.066
0.08	749	0.546	1067	0.066
0.09	661	0.546	990	0.083
0.10	641	0.546	914	0.10

ESPRIT is evaluated on Needleman-Wunsch (NW) global alignment distance after using k-mer distance as input. LSH-Div is evaluated on ESPRIT's generated k-mer distance and NW distance. LSH-Div parameter setting for this experiment is k = 30, w-mer = 3.

VI. Conclusion

We presented a scalable species diversity estimation algorithm LSH-Div for metagenomic samples. The algorithm utilizes an efficient randomized locality sensitive hash function to approximate the pairwise sequence similarity procedure.



(a) Number of OTUs (b) Chao1 Index (c) Ace Index

Figure 3: Species Richness Estimation for 53R sample.
Number of OTUs, Chao1 Index and ACE Index produced by LSH-Div at different distance cut-offs. LSH-Div parameter setting for this experiment is k = 30, w-mer = 3.

The algorithm is enriched to use gapless, subsequences of fixed length (w-mer) which contributes to a reduction in number of false positives, computational speedup and improvement in OTU estimation results. We evaluate LSH-Div on eight environmental samples and performed a comprehensive study of the different parameters and their impact on OTU estimation, diversity indexes and and computational time. The code is written in Python and is made available publicly under the GNU GPL license. We are extending the LSH-Div methodology to handle whole metagenome sequences.

References

[1] J. Venter *et al.*, "Environmental genome shotgun sequencing of the Sargasso Sea," *Science*, vol. 304, no. 5667, p. 66, 2004.

[2] P. Hugenholtz *et al.*, "Microbiology: metagenomics." *Nature*, vol. 455, no. 7212, pp. 481–3, Sep. 2008.

[3] S. M. Huse *et al.*, "Exploring microbial diversity and taxonomy using ssu rrna hypervariable tag sequencing," *PLoS Genet*, vol. 4, no. 11, p. e1000255, 11 2008.

[4] Z. Liu *et al.*, "Accurate taxonomy assignments from 16S rRNA sequences produced by highly parallel pyrosequencers." *Nucleic acids research*, vol. 36, no. 18, p. e120, Oct. 2008.

[5] Q. Wang *et al.*, "Naïve Bayesian Classifier for Rapid Assignment of rRNA Sequences into the New Bacterial Taxonomy," *Applied and Environmental Microbiology*, vol. 73, no. 16, pp. 5261–5267, Aug. 2007.

[6] D. Schloss, Patrick *et al.*, "Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities," *Appl. Environ. Microbiol.*, vol. 75, no. 23, pp. 7537–7541, 2009.

[7] P. D. Schloss *et al.*, "Introducing dotur, a computer program for defining operational taxonomic units and estimating species richness," *Appl. Environ. Microbiol.*, vol. 71, no. 3, pp. 1501–1506, 2005.

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

[9] P. D. Schloss *et al.*, "Assessing and Improving Methods Used in Operational Taxonomic Unit-Based Approaches for 16S rRNA Gene Sequence Analysis," *Appl. Environ. Microbiol.*, vol. 77, no. 10, pp. 3219–3226, May 2011.

[10] J. B. Hughes *et al.*, "Counting the uncountable: statistical approaches to estimating microbial diversity," vol. 67, no. 10, pp. 4399–406+, 2001.

[11] P. Indyk *et al.*, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, ser. STOC '98. New York, NY, USA: ACM, 1998, pp. 604–613.

[12] J. Buhler, "Efficient large-scale sequence comparison by locality-sensitive hashing," *Bioinformatics*, vol. 17, no. 5, pp. 419–428, 2001.

[13] Z. Rasheed *et al.*, "Efficient clustering of metagenomic sequences using locality sensitive hashing," *SIAM International Conference on Data Mining (SDM12)*, pp. 1023–1034, 2012.

[14] M. L. Sogin *et al.*, "Microbial diversity in the deep sea and the underexplored rare biosphere," *Proceedings of the National Academy of Sciences*, vol. 103, no. 32, pp. 12 115–12 120, 2006.

[15] S. Huse *et al.*, "Accuracy and quality of massively parallel dna pyrosequencing," *Genome Biology*, vol. 8, no. 7, p. R143, 2007.

[16] M. Margulies *et al.*, "Genome sequencing in microfabricated high-density picolitre reactors," *Nature*, vol. 437, no. 7057, pp. 376–380, Jul. 2005.

[17] A. Chao, "Nonparametric Estimation of the Number of Classes in a Population," *Scandinavian Journal of Statistics*, vol. 11, no. 4, 1984.

[18] C. J. Krebs, *Ecological Methodology.* New York.: Harper & Row, 1989.

[19] A. Chao *et al.*, "Estimating the Number of Classes via Sample Coverage," *Journal of the American Statistical Association*, vol. 87, no. 417, pp. 210–217, 1992.

[20] P. D. Schloss, "The effects of alignment quality, distance calculation method, sequence filtering, and region on the analysis of 16s rrna gene-based studies," *PLoS Comput Biol*, vol. 6, no. 7, p. e1000844, 07 2010.