ESPRIT: Estimating Species Richness Using Large Collections of 16S rRNA Shotgun Sequences (Supplementary Data)

Yijun Sun[§], Yunpeng Cai[¶], Li Liu[§], Fahong Yu[§]
Michael L. Farrell[♯], William McKendree[♯], William Farmerie[§]

[§]Interdisciplinary Center for Biotechnology Research
[¶]Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32610-3622
[♯]Materials Technology Directorate
Air Force Technical Applications Center
1030 S. Highway A1A, Patrick AFB, FL 32925-3002

1 Pairwise or Multiple Sequence Alignment?

In contrast to many existing metagenomics studies, we used pairwise sequence alignment, instead of multiple sequence alignment (MSA), to align 16S rRNA sequences. This is a critical issue that merits more explanations. MSA is used to infer homology segments of input sequences. An underlying assumption is that input sequences should share some similarities, which may not be valid for 16S rRNA based studies that target on hypervariable regions of rRNA gene (e.g., V6 and V3 regions). It has been reported in the literature that microbial communities are much more diverse than expected [1, 2, 3]. To further demonstrate this, we performed a simulation study where we calculated the pairwise distances of the sequences of a seawater sample and plotted a histogram of the distances in Figure 1. We report that only about 2% sequence pairs have a genetic distance smaller than 0.1. In other words, for every two randomly selected sequences, there is a 98% probability that the two sequences are from two distantly related OTUs. The sequences were aligned by using the Needleman-Wunsch algorithm. We will shortly see that if MSA is used, the histogram will be further skewed toward the right. We then performed another simulation study to demonstrate how the presence of a large proportion of highly diverse sequences affects the alignment of sequences with a small genetic distance. The following describes

in detail how the experiment was conducted. We first randomly selected two sequences with a distance of 0.06 calculated based on pairwise alignment, then performed a multiple sequence alignment of the two sequences along with 100 sequences randomly selected from the data (using MUSCLE with the default parameter), and recorded the pairwise distance of the first two sequences. In order to eliminate statistical variations, the experiment was repeated 100 times. The so-obtained distances are plotted in Figure 2, from which we can see that the pairwise distances computed based on MSA are much larger than 0.06. The maximum distance is 0.22 and the average is 0.09. This can be explained by the fact that MSA is aimed to minimize the sum of pairwise scores, and in order to align the 100 highly diverse sequences, the alignment quality of the first two sequences has to be sacrificed, leading to an inflated estimate of genetic distance. Note that we include only 100 randomly selected sequences in this experiment. With more unrelated sequences and/or if unrelated sequences are more diverse, the distance between the first two sequences would be even larger. This leads to an interesting observation: whether two sequences are from the same OTU (e.g., species or genus) should be physically determined by their sequence composition. Now with the use of MSA, the distance and thus whether the two sequences belong to the same OTU are also determined by sequencing depth (i.e., the number of sequences sampled) and the environment from which they are extracted (i.e., how diverse other sequences are), which is a highly undesirable property. Another interesting thing we should point out is that the harder an MSA algorithm tries to minimize the sum of pairwise scores by aligning highly diverse sequences, the larger the distance between the two first sequence can be. This is exactly the reason why MUSCLE with the default parameter yields a worse result than that obtained by using the simple parameter (Figure 3 in the main text). From the above analysis, we conclude that MSA should not be used for the purposes of estimating microbial diversity, even if the optimal MSA could be performed.

2 NAST and RDP-Pyro

We performed some experiments to compare the prediction performance of ESPRIT with those obtained by using NAST [6] and Pyro [5]. Speaking strictly, NAST is not a computational algorithm for taxonomy independent analysis, but a multiple sequence aligner. We replaced the Needleman-Wunsch algorithm with NAST in our ESPRIT algorithm. Unlike MUSCLE where input sequences are compared against each other, NAST performs the alignment by comparing input sequences against a set of core sequences. As with all

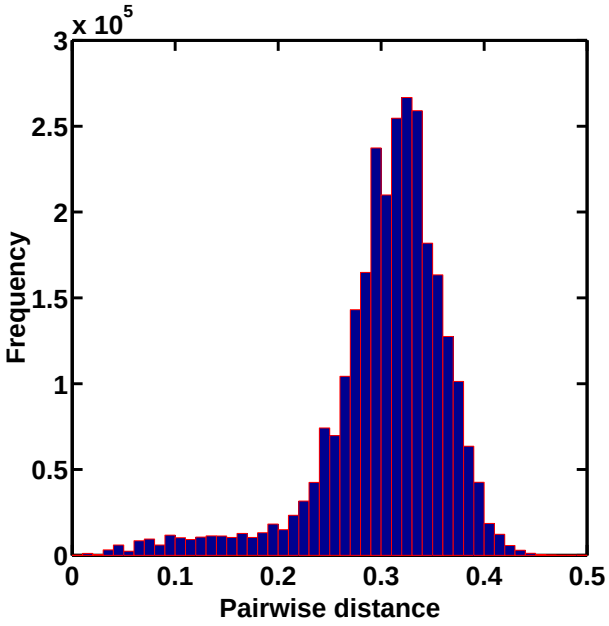

Figure 1: Sequence pairs with distances less than 0.10 only account for a very small fraction of all possible pairs (2.25% for this example). The experiment was performed on the 53R seawater sample.

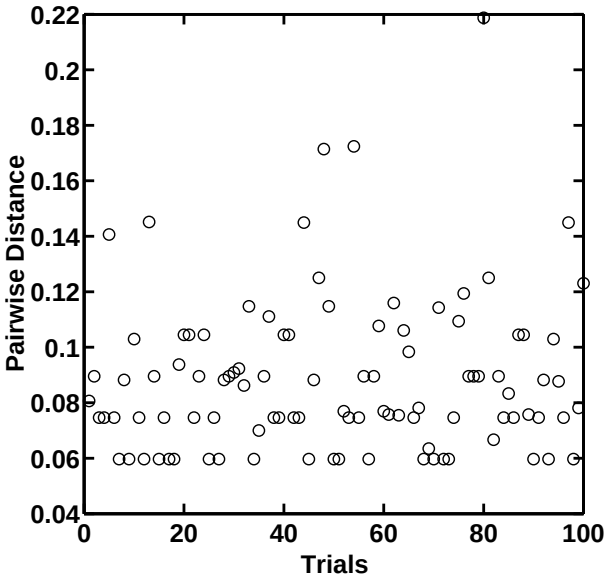

Figure 2: Pairwise distances computed based on multiple sequence alignment are much larger than that obtained by using the Needleman-Wunsch algorithm, which is 0.06 in this example. The maximum distance is 0.22 and the average is 0.09.

other MSA algorithms, NAST is computationally expensive and can process only about 10 sequences per minutes [6]. Moreover, NAST is a web application that only allows at most 500 sequences per batch [6]. For a large project, one has to break the data into some small pieces. Pyro is a recently developed 16S rRNA data analysis pipeline as a part of the well-known RPD project. Pyro is fundamentally different from all other existing methods: it first trains computational models using a small number of representative rRNA sequences, and then aligns each query sequence against trained models (instead of other sequences). Hence, the alignment module of Pyro has a linear (instead of quadratic) computational complexity with respect to the number of sequences, and has the potential to scale to large projects. However, as with all other model-based methods, the performance of Pyro relies on how accurate the trained models represent a diverse microbial community. As with NAST, Pyro is also a web application that allows users to submit sequences through the Internet. Pyro can process a maximum of slightly more than 150K unique sequences, due to the limitation of the clustering tool used in the algorithm [5].

Figures 3 and 4 present the numbers of OTUs as a function of distance levels, estimated by various algorithms performed on the two benchmark datasets. Since NAST is a MSA algorithm, the estimation results are consistent with those presented in the previous section. We did not spend extra efforts to investigate why NAST performs worse than MUSCLE. One possible reason is that NAST yields a better overall alignment of input sequences measured by the sum of pairwise scores, leading to larger genetic distances among closely related sequences, compared to MUSCLE using the simple parameter. It is interesting to note that the numbers of OTUs estimated by Pyro are much smaller than those obtained by all other methods when the pairwise distance is small, but larger than MUSCLE and ESPRIT when the distance level is relatively large. We should emphasize that the number of OTUs does not tell the whole story; what we are really interested in is how individual sequences are grouped into OTUs. We found that Pyro assigns a distance of zero to a large number of seemingly unrelated sequences. Since Pyro is a web application and the paper does not provide a detailed description of the algorithm and numeric examples, we send these sequences to the RDP staff. They were very helpful and confirmed that this issue lies in the alignment where only *one* model position is shared by these sequences. When a sequence is not represented by any model but has to find a hit, the above error will occur. This is exactly the reason why the number of OTUs estimated by Pyro is smaller than all other methods when the sequence variation is small. As mentioned before, microbial communities are very diverse. For example, it is estimated by [12, 3] that the number of

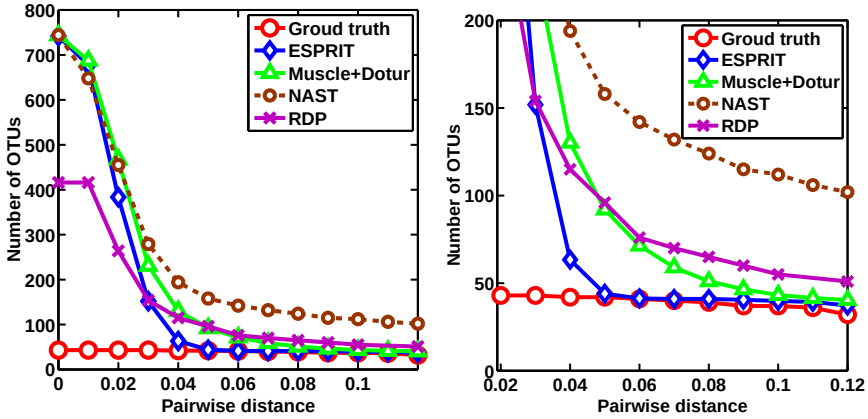

Figure 3: Numbers of OTUs as a function of distance levels estimated by using ESPRIT, MUSCLE+DOTUR, NAST, and Pyro algorithms performed on simulation data with each read containing up to 3% sequencing errors. The right panel is the zoom-in figure of the left panel.

species of bacterial per gram of soil varies between 2000 to 8.3 million. Though this number is debatable, it might not be possible to accurately represent such diverse environments using a limited number of models. This problem is not limited to Pyro, but exists in other model-based approaches. It is possible to improve the prediction accuracy of Pyro by using more rRNA representative sequences for training, but in the current formulation, it seems that Pyro performs worse than a model-free approach such as ESPRIT.

3 Preparation of Air Sample

Air filters containing particulate from over 100,000 m^3 of filtered air were extracted for DNA. The DNA was quantified by Qubit (Invitrogen) using the Quant-iT dsDNA HS assay kit. One ng of DNA was added to each of 96 wells containing 20 μl of PCR master-mix consisting of BioRad's SYBR Green SuperMix and 500 nM equimolar ratio of primer adapter Am 5'- CCATCTCATCCCTGCGTGTCCCATCTGTTCCCTCCCTGTCTCAG (GITACCTTGTTACGACTT) -3', and biotinylated primer adapter Bm 5'- BiotinTEG/ CCTATCCCCTGTGTGCCTTGCCTATCCCCTGTTGCGTGTCTCAG (ATTAGATAC-CCIGGTAG) -3'. The internal 16S ribosomal RNA primer sequences (in parentheses) of the Am and Bm adapter primers were the modified 1492r and 787f sequences, respectively, adopted from [9] and target the 16S rRNA V6 hypervariable region. PCR reactions were

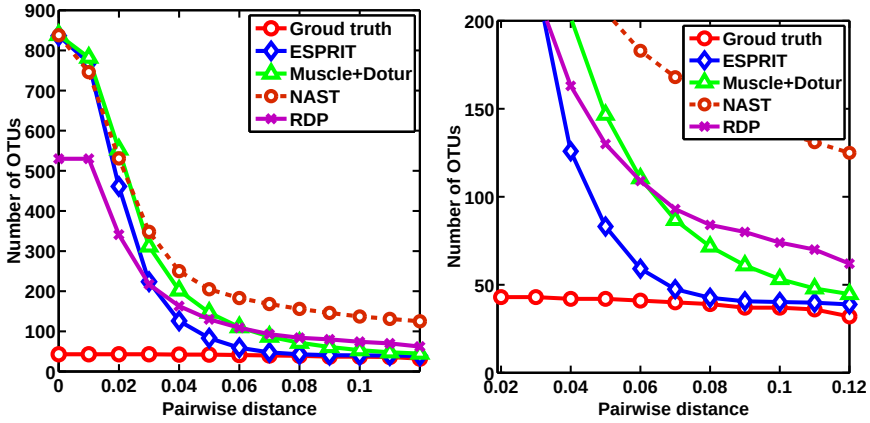

Figure 4: Numbers of OTUs as a function of distance levels estimated by using ESPRIT, MUSCLE+DOTUR, NAST, and Pyro algorithms performed on simulation data with each read containing up to 5% sequencing errors. The right panel is the zoom-in figure of the left panel.

subjected to melting at 94 °C for 2 min followed by 25 cycles of 94 °C denaturing for 45 sec, 55 °C annealing for 45 sec, and 72 °C extension for 1 min. After 25 cycles, the reactions were subjected to additional 72 °C extension for 6 min. Residual primers and unincorporated NTPs were removed by Montage PCR filter units (Millipore). Cleaned elutions were pooled, quantified as described above, and visualized on a gel to confirm a discreet 705 bp product. The prepared 16S PCR product material was subjected to the emPCR process to make the bead-attached single stranded templates required for 454 pyrosequencing as described by [8].

4 Hcluster Algorithm

Hcluster is a hierarchical clustering algorithm. Conventional methods take a full distance matrix as input. They work well when the number of sequences is small. However, given a full run of 454 data, a full distance matrix can be as large as 1000GB. Consequently, conventional methods can handle only a limited number of sequences. In contrast, Hcluster uses a sparse matrix and processes distances *on-the-fly*, thus overcoming the memory issue and paving the way for processing millions of reads. The idea is very simple, but it took us considerable efforts to optimize the code. Hcluster currently only supports complete linkage. It has been shown in [12] that complete linkage gives a more conservative result

than average/nearest linkages for the purposes of estimating microbial diversity. However, Hcluster can be easily extended to support average/nearest linkages. We below provide a toy example to illustrate how it works.

Let us consider a distance matrix generated from five sequences (Table 1). We first transform it into a sparse matrix where the distances are sorted in an ascending order, as shown in Table 2.

A (1)	0.00	0.01	0.015	0.03	0.04
B (2)	0.01	0.00	0.03	0.02	0.05
C (3)	0.015	0.03	0.00	0.02	0.03
D (4)	0.03	0.02	0.02	0.00	0.02
E (5)	0.04	0.05	0.03	0.02	0.00

Table 1: Distance matrix of a toy example

Row	Col	Value
1	2	0.01
1	3	0.015
3	4	0.02
2	4	0.02
4	5	0.02
2	3	0.03
1	4	0.03
3	5	0.03
1	5	0.04
2	5	0.05

Table 2: Sparse matrix

In order to perform clustering sequentially, we introduce a cluster array to record the state information of each cluster generated during hierarchical clustering, and a link table to record the number of links between active clusters. The state information includes the number of sequences (denoted by the Num_Seqs field) in a cluster and the index of its topmost parent cluster (denoted by the $Parent$ field. If no parent exists, the field is set to -1). The link table contains an index array storing active clusters, and a symmetry matrix that counts the number of links between the active clusters that have been processed.

Initially, the link table is empty and each sequence is regarded as a single cluster with no parent. Hence, the first five unit of the cluster array is set to 1 in the Num_Seqs field and -1 in the $Parent$ field:

Cluster Array

Index	1	2	3	4	5	0	0	0	0
Num_Seqs	1	1	1	1	1				
$Parent$	-1	-1	-1	-1	-1				

Link Table

*		
*	*	
*	*	*

Now we begin to process the distance matrix presented in Table 2 step by step. The following operations are repeated until the entire matrix is processed:

1. Read a pairwise distance;
2. Update the cluster array and the link table;
3. If two clusters can be merged into one, merge them;
4. If cluster merging takes place, update the cluster array and the link table;

The first input is $(1, 2, 0.01)$. Here, the first two elements represent the indexes of two sequences and the third one is the distance between them. We look up the link table and recognize that neither cluster 1 nor cluster 2 was active in the previous steps. With the new input, they become active and are added to the link table:

Cluster Array

Index	1	2	3	4	5	0	0	0	0
Num_Seqs	1	1	1	1	1				
$Parent$	-1	-1	-1	-1	-1				

Link Table

	1	**2**	
1	*	**1**	
2	*	*	
	*	*	*

Some explanations on the link table are necessary. We use $L(a, b)$ to denote the number of the links between clusters a and b stored in the link table. Since the matrix is symmetry, $L(a, b) = L(b, a)$. We skip the trivial details of how to access $L(a, b)$ in the matrix. By adding clusters 1 and 2, we set $L(1, 2)$ to 1.

We then check if cluster merging is needed. For clusters a and b containing $Num\_Seqs(a)$ and $Num\_Seqs(b)$ sequences, respectively, the maximum possible number of links between a and b is $l\_max = Num\_Seqs(a) \times Num\_Seqs(b)$. If $L(a,b) = l\_max$, then the complete linkage information between a and b is known. Since the input is sorted in an ascending order and in each step only one pair of clusters is updated, we only need to examine whether the current pair or their parents meet this criterion. If yes, they are guaranteed to be the pair that has the minimal complete linkage distance among all unmerged clusters, and thus can be merged together immediately.

In the above example, $Num\_Seqs(1) \times Num\_Seqs(2) = 1$ and $L(1,2) = 1$. Thus, clusters 1 and 2 can be merged into one cluster. The cluster array is then modified by adding a new cluster with index 6, and the *Parent* of clusters 1 and 2 are pointed to it. Clusters 1 and 2 in the link table are also merged. The link information of both clusters is passed to cluster 6.

Cluster Array

Index	1	2	3	4	5	**6**	0	0	0
Num_Seqs	1	1	1	1	1	**2**			
Parent	**6**	**6**	-1	-1	-1	-1			

Link Table

	6		
6	*		
	*	*	
	*	*	*

Before proceeding, we describe the rule of merging the link information of two clusters in the link table. Suppose that clusters a and b form a new cluster c. The following operations are performed:

$$L(c,x) \leftarrow L(x,a) + L(x,b) \qquad \forall x \neq a \;\&\; x \neq b \;,$$
$$L(x,a) \leftarrow 0 \;,\; L(x,b) \leftarrow 0 \qquad \forall x \;.$$

Then, a and b are removed from the link table, and $L(a,x)$ and $L(b,x)$ are no longer valid.

The clustering result is reported to the output:

Output

1	**2**	**0.01**	$->$	**6**

Now we go to the next input $(1, 3, 0.015)$. Because the topmost parent of sequence 1 is 6, we examine the pair $(6, 3)$ instead. Cluster 3 becomes an active cluster and is added to the link table with $L(6,3)$ set to 1:

Cluster Array

Index	1	2	3	4	5	6	0	0	0
Num_Seqs	1	1	1	1	1	**2**			
$Parent$	6	6	-1	-1	-1	-1			

Link Table

	6	**3**	
6	$*$	**1**	
3	$*$	$*$	
	$*$	$*$	$*$

Since $L(6,3) < Num\_Seqs(6) \times Num\_Seqs(3)$, no cluster merging happens and we proceed to the next input $(3, 4, 0.02)$. The link table is updated using the previous described rules:

Cluster Array

Index	1	2	3	4	5	6	0	0	0
Num_Seqs	1	1	1	1	1	**2**			
$Parent$	6	6	-1	-1	-1	-1			

Link Table

	6	3	**4**
6	$*$	1	
3	$*$	$*$	**1**
4	$*$	$*$	$*$

Now sequences 3 and 4 are merged into one cluster with a new index 7. By following the rules described above, the link information $L(3,6)$ is passed to $L(6,7)$. Both the cluster array and link table are changed into:

Cluster Array

Index	1	2	3	4	5	6	**7**	0	0
Num_Seqs	1	1	1	1	1	2	**2**		
$Parent$	6	6	**7**	**7**	-1	-1	-1		

Link Table

	6	**7**	
6	$*$	1	
7	$*$	$*$	
	$*$	$*$	$*$

and a new line is added to the output:

Output

1	2	0.01	$->$	6
3	**4**	**0.02**	$->$	**7**

We proceed to the next input $(2, 4, 0.02)$. Since the topmost parents of clusters 2 and 4 are clusters 6 and 7, respectively, we perform the operation $L(6,7) \leftarrow L(6,7) + 1$ and nothing else needs to be changed. The similar operations are performed for the next three inputs $(4, 5, 0.02)$, $(2, 3, 0.03)$ and $(1, 4, 0.03)$. The cluster array remains unchanged while the link table is modified into:

Link Table			
	6	7	**5**
6	*	**4**	
7	*	*	**1**
5	*	*	*

Clusters 6 and 7 now have a complete link since $L(6,7) = Num\_Seqs(6) \times Num\_Seqs(7)$, and thus can be merged into a new cluster 8. Accordingly, all clusters whose topmost parents were assigned to clusters 6 or 7 should be modified, as shown below:

Cluster Array									
Index	1	2	3	4	5	6	7	**8**	0
Num_Seqs	1	1	1	1	1	2	2	**4**	
Parent	**8**	**8**	**8**	**8**	-1	**8**	**8**	-1	

Link Table			
	8	5	
8	*	1	
5	*	*	
	*	*	*

The output is printed:

Output				
1	2	0.01	$->$	6
3	4	0.02	$->$	7
6	**7**	0.03	$->$	**8**

We skip the remaining steps. Finally, we get the following clustering result:

Output				
1	2	0.01	$->$	6
3	4	0.02	$->$	7
6	7	0.03	$->$	8
8	5	0.05	$->$	9

The clustering diagram is shown in Figure 5.

From the above toy example, we can see that Hcluster demands only a link table that is much smaller in size than a full distance matrix. When applied to real data, the link table is usually very sparse, and thus the memory required can be further reduced significantly. Although an additional cluster array is required, its size grows only linearly with respect to the number of sequences, which is negligible when many sequences are processed. For these reasons, Hcluster is capable of handling many hundreds of thousands of sequences in a personal computer.

4.1 Benchmark Study

The accuracy of Hcluster was benchmarked against DOTUR and the latest version of MOTHUR (`http://schloss.micro.umass.edu/mothur/Main_Page`). The same distance matrix generated by using the Needleman-Wunsch algorithm was used in the three algorithms to facilitate the comparison. The performance of both DOTUR and MOTHUR is dependent on the choice of precision level (i.e., the -p flag in DOTUR and the precision parameter in MOTHUR). Using a precision level equal to that of the distance matrix, which is 10000 in this experiment, yields an exact solution, while using a lower precision level can significantly reduce computational time of the two algorithms but results in only an approximate solution. By setting the precision level to 10000, the three methods generated the exactly same result (Figure 6). This is expected since the three algorithms implement the standard hierarchical clustering. We also performed an experiment to compare the computational complexity of the three algorithms. The parameter -stop in DOTUR and the cutoff value in MOTHUR were set to 0.10, and the precision level was 100. Hence, the comparison is somehow in favor of DOTUR and MOTHUR. When applying MOTHUR, we removed the duplicated sequences and used a sparse matrix that contains only the sequence pairs with distances less than the cutoff value, which significantly reduced the processing time and the size of the resulting distance matrix. We applied the three algorithms to the FS396 dataset, which contains 17666 sequence and is the largest dataset in size among the eight seawater samples. The running time is reported in Table 3. Since DOTUR integrated both clustering and statistical analysis into one program, for fairness of comparison, the running time of both MOTHUR and ESPRIT spent on statistical inference is also reported. As can be seen from the table, Hcluster is computationally very efficient and takes about 4 and 7 seconds to cluster and statistically analyze the FS396 data, respectively. In contrast, it takes DOTUR about 128 minutes to finish the two tasks. MOTHUR significantly improves DOTUR, but performs slightly worse than HCluster. As with DOTUR, MOTHUR loads a distance matrix into the memory before proceeding to perform clustering. Hence, it does not fundamentally address the computational issue associated with processing massive pyrosequencing data. As mentioned in the main text, given a full run of 454 data, a full distance matrix can be as large as 1000 GB. Even if we remove duplicated sequences and sequence pairs that have a pairwise distance larger than a cutoff value (say 0.1), the resulting distance matrix in a sparse format can be 20GB, which is too big to be directly loaded into the memory in most computers. For this reason, we were unable to process the air sample using DOTUR and MOTHUR. It should be noted that the size of a distance

Table 3: Comparison of the running times of ESPRIT, MOTHUR, and DOTUR spent on clustering and statistical analysis.

		ESPRIT		MOTHUR		
data	Num. of reads	cluster	statistical analysis	cluster	statistical analysis	DOTUR
FS396	17666	4s	7s	52s	67s	128m

matrix grows quadratically with respect to the number of sampled sequences. Hcluster addresses this problem by processing distances on-the-fly (i.e., process one distance at a time).

References

[1] Sogin, M.L., Morrison, H.G., Huber, J.A., Welch, D.M., Huse, S.M., Neal, P.R., Arrieta, J.M. and Herndl, G.J. (2006) Microbial diversity in the deep sea and the underexplored "rare biosphere". *Proc. Natl. Acad. Sci. USA*, **103**, 12115–12120.

[2] Keijser, B., Zaura, E., Huse, S.M., van der Vossen, J., Schuren, F., Montijn, R.C., ten Cate, J.M. and Crielaard, W. (2008) Pyrosequencing analysis of the oral microflora of healthy adults. *J. Dent. Res.*, **87**, 1016–1020.

[3] Gans, J., Woilinsky, M. and Dunbar, J. (2005) Computational improvements reveal great bacterial diversity and high metal toxicity in soil. *Science*, **309**, 1387–1390.

[4] Schloss, P.D., Handelsman, J. (2006) Toward a census of bacteria in soil. *PLOS Computational Biology*, **2**, 786793.

[5] Cole, J.R., Wang, Q., Cardenas, E., Fish, J., Chai, B., Farris, R.J., Kulam-Syed-Mohideen, A.S., McGarrell, D.M., Marsh, T., Garrity, G.M. and Tiedje J.M. (2009) The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic Acids Res.*, **37**, D141–145.

[6] DeSantis, T.Z., Hugenholtz, P., Keller, K., Brodie, E.L., Larsen, N., Piceno, Y.M., Phan, R., Andersen, G.L. (2006) NAST: a multiple sequence alignment server for comparative analysis of 16S rRNA genes. *Nucleic Acids Res.*, **34**, W394–399.

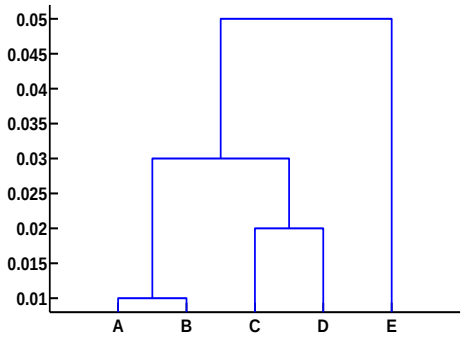

Figure 5: Clustering result of the toy example.

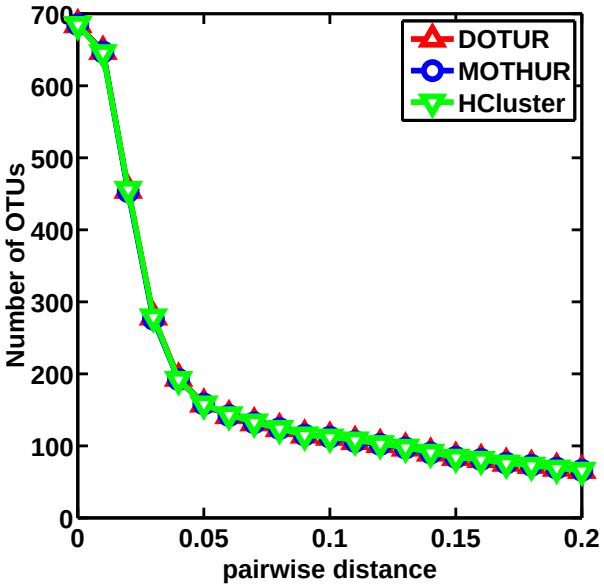

Figure 6: The accuracy of Hcluster is benchmarked against DOTUR and MOTHUR. The three methods yield the exactly same result.

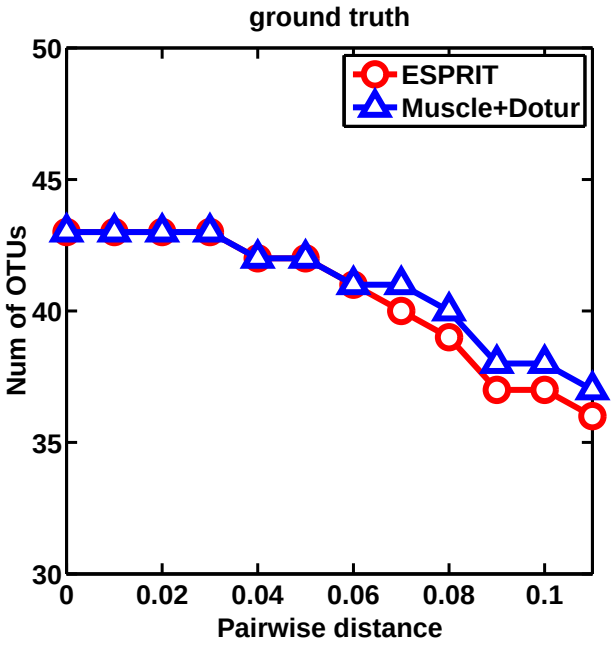

Figure 7: Lineage-through-time curves of the 43 reference 16S rRNA genes, generated by using ESPRIT and MUSCLE+Dotur algorithms. They served as the ground truth in the benchmark study.

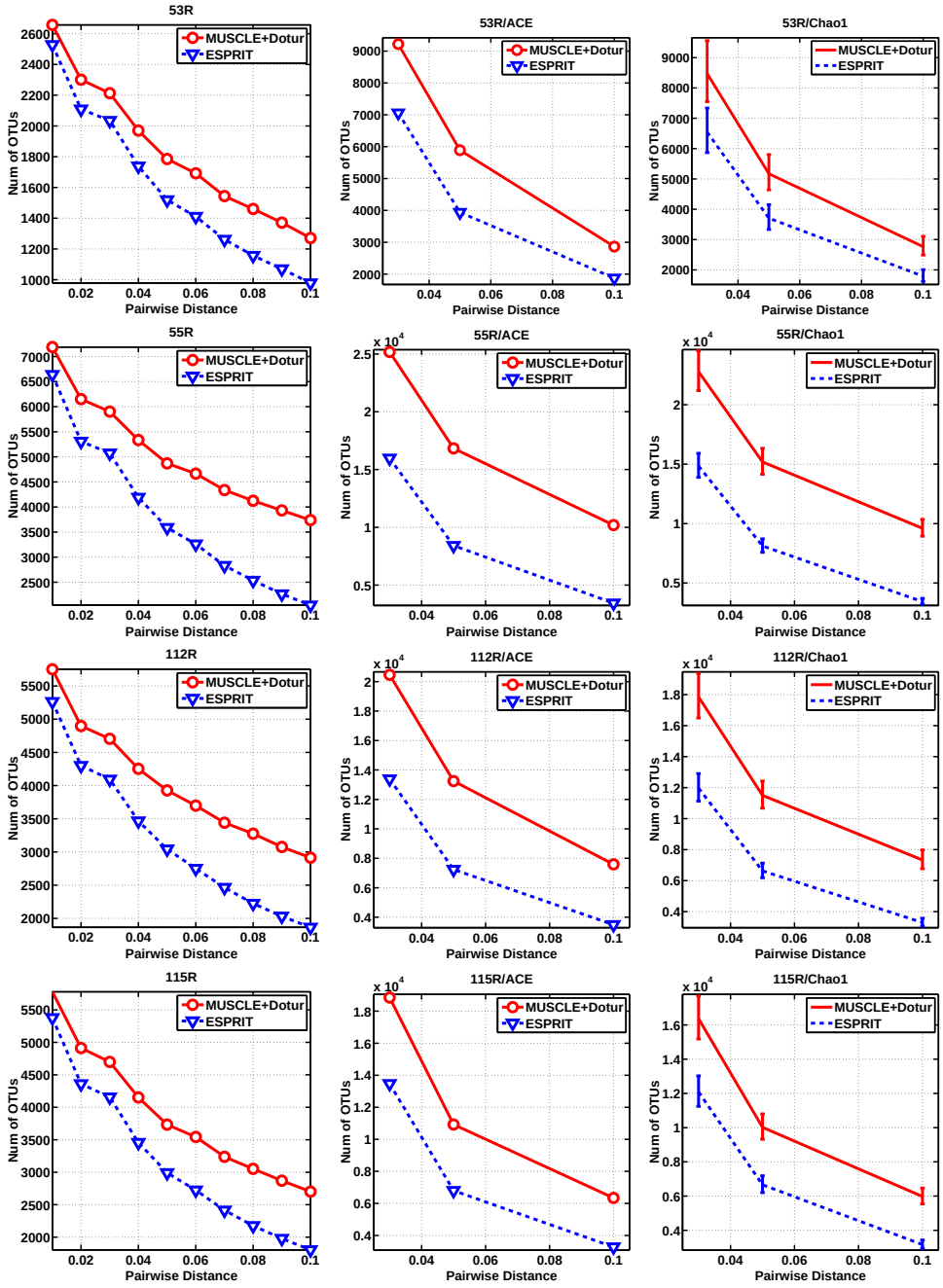

Figure 8: Lineage-through-time curves, ACE and Chao1 estimates generated by using ESPRIT and MUSCLE+Dotur algorithms performed on 53R, 55R, 112R and 115R datasets.

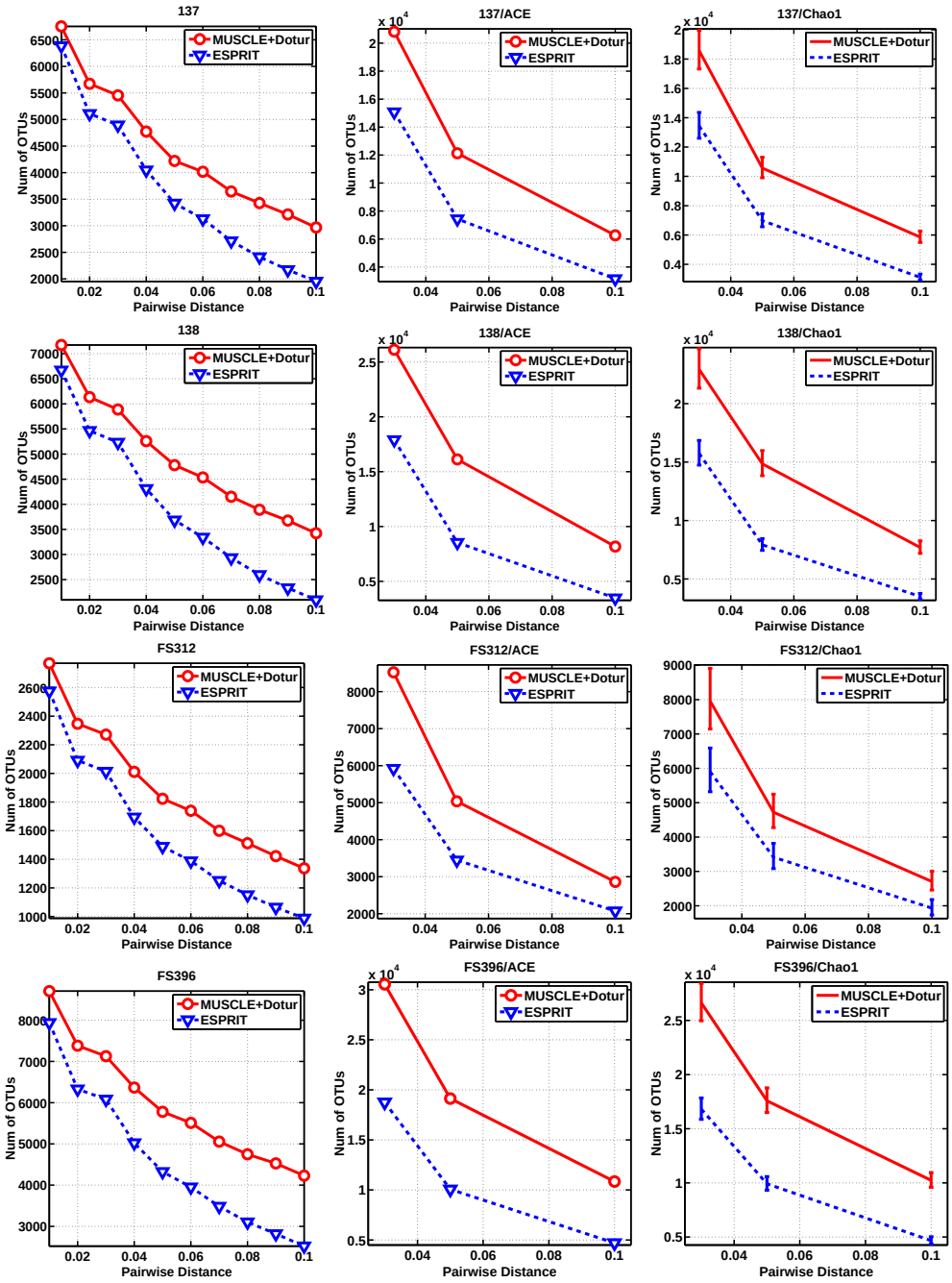

Figure 9: Lineage-through-time curves, ACE and Chao1 estimates generated by using ESPRIT and MUSCLE+Dotur algorithms performed on 137, 138, FS312 and FS396 datasets.

[7] Polz, M.F., and Cavanaugh, C.M. (1998) Bias in template-to-product ratios in multi-template PCR. *Appl. Environ. Microbiol.*, **64**:3724-3730.

[8] Margulies, M., Egholm, M., Altman, W.E., Attiya, S., Bader, J.S., Bemben, L.A., Berka, J., Braver man, M.S., Chen, Y.J., Chen, Z., et al. (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.

[9] Roesch, L.F.W., Fulthorpe, R.R., Riva, A., Casella, G., Hadwin, A.K.M., Kent, A.D., Daroub, S.H., Camargo, F.A.O., Farmerie, W.G. and Triplett, E.W. (2007) Pyrosequencing enumerates and contrasts soil microbial diversity. *The ISME Journal*, **1**, 283–290.

[10] Berkhin, P. (2006) A survey of clustering data mining techniques, *Grouping Multidimensional Data*, Springer Berlin Heidelberg.

[11] Duda, R.O., Hart, P.E. and Stork, D.G. (2001) Pattern Classification (2nd edition). Wiley, New York.

[12] Schloss, P.D. and Handelsman, J. (2005) Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. *Appl. Environ. Microbiol.*, **71**, 1501–1506.