The Journal of Microbiology (2012) Vol. 50, No. 2, pp. 181–185
DOI 10.1007/s12275-012-1214-6

TBC: A Clustering Algorithm Based on Prokaryotic Taxonomy[§]

Jae-Hak Lee[1], Hana Yi[2], Yoon-Seong Jeon[1,5], Sungho Won[3], and Jongsik Chun[1,2,4,5]*

[1]Interdisciplinary Graduate Program in Bioinformatics, [2]Inst. of Molecular Biology and Genetics, Seoul National University, Seoul 151-742, Republic of Korea
[3]Department of Statistics, Chung-Ang University, Seoul 156-756, Republic of Korea
[4]School of Biological Sciences and Advanced Inst. of Convergence Tech., Seoul National University, Seoul 151-742, Republic of Korea
[5]Chunlab, Inc., Bldg 138 Rm 318, Seoul National University, Seoul 151-742, Republic of Korea

(Received April 25, 2011 / Accepted November 7, 2011)

High-throughput DNA sequencing technologies have revolutionized the study of microbial ecology. Massive sequencing of PCR amplicons of the 16S rRNA gene has been widely used to understand the microbial community structure of a variety of environmental samples. The resulting sequencing reads are clustered into operational taxonomic units that are then used to calculate various statistical indices that represent the degree of species diversity in a given sample. Several algorithms have been developed to perform this task, but they tend to produce different outcomes. Herein, we propose a novel sequence clustering algorithm, namely Taxonomy-Based Clustering (TBC). This algorithm incorporates the basic concept of prokaryotic taxonomy in which only comparisons to the type strain are made and used to form species while omitting full-scale multiple sequence alignment. The clustering quality of the proposed method was compared with those of MOTHUR, BLASTClust, ESPRIT-Tree, CD-HIT, and UCLUST. A comprehensive comparison using three different experimental datasets produced by pyrosequencing demonstrated that the clustering obtained using TBC is comparable to those obtained using MOTHUR and ESPRIT-Tree and is computationally efficient. The program was written in JAVA and is available from http://sw.ezbiocloud.net/tbc.

Keywords: TBC, clustering algorithm, OTU, CD-HIT, UCLUST, MOTHUR, ESPRIT-Tree, BLASTClust, pyrosequencing, metagenome

Introduction

Thanks to recent advancements in DNA sequencing technology, methods for the elucidation of microbial community structures have shifted from indirect methods, such as DGGE, t-RFLP, and DNA microarrays, to direct methods, that is, the sequencing of amplified phylogenetic marker genes. The Roche GS FLX Titanium system, for example, can generate one million sequencing reads of 400–500 bp in length per run (Metzker, 2010). Massive sequencing of PCR amplicons targeting the 16S rRNA gene has been widely used to dissect the microbial community structure of a variety of environmental samples. For example, the microbial inhabitants of the human body and natural environments have been successfully surveyed using massive pyrosequencing of the 16S rRNA genes amplified from metagenomic DNA (Petrosino *et al.*, 2009).

Following massive gene sequencing, fundamentally important ecological aspects of microbial diversity are calculated. "Alpha" diversity refers to the diversity within a sample/community, whereas "beta" diversity is defined as the diversity among multiple samples/communities, thus reflecting how samples are related (Hamady and Knight, 2009). Measuring these diversity-related indices has been considered essential in microbial community analysis and ecological studies. There are two methods for calculating diversity indexes, namely taxon-based and phylogeny-based approaches. The first approach calculates diversity indices by clustering individual sequences into a group, named an operational taxonomic unit (OTU), and the latter does so by considering the phylogenetic relationships of each sequence. Although a phylogeny-based approach may be more informative in an evolutionary context, due to the phylogeny-based approach's complexity, the taxon-based approach is more widely used.

Taxon-based estimation of "alpha" diversity can be divided into two separate steps. The first step involves the clustering of sequences into OTUs. The resultant OTUs with their members (sequences) can then be used to calculate various non-parametric and parametric diversity indexes, including Chao1 (Chao, 1984) and ACE (Chao and Lee, 1992; Chao *et al.*, 1993).

The clustering of multiple sequences is generally achieved by multiple sequence alignment (MSA), followed by the calculation of distance matrices (Bacon and Anderson, 1986). The most popular algorithms for this approach are the nearest-neighbor clustering and furthest-neighbor clustering algorithms implemented in the MOTHUR program (Schloss *et al.*, 2009), which uses a distance matrix generated from the MSA as an input file. Distance matrices can be generated by various computer programs, such as ClustalW (Thompson *et al.*, 1994) and MUSCLE (Edgar, 2004). Although this approach has a sound statistical basis, it has a high computational cost. In fact, MSA-based clustering is not practical for the analysis of the large number of sequences generated by Roche 454 sequencing.

*For correspondence. E-mail: jchun@snu.ac.kr Tel.: +82-2-880-8153 Fax: +82-2-874-8153
[§]Supplemental material for this article may be found at http://www.springer.com/content/120956

Therefore, MSA-free clustering algorithms such as BLASTClust (Altschul *et al.*, 1997), ESPRIT-Tree (Cai and Sun, 2011), CD-HIT (Li and Godzik, 2006), and UCLUST (Edgar, 2010) have been introduced and have become widely used. BLASTClust employs the popular BLAST program to identify similar sequences that are then used to form a cluster. ESPRIT-Tree partitions an input sequence space into a set of subspaces using a partition tree that is constructed using a pseudo-metric and then recursively refines the clustering structure in these subspaces. To avoid exhaustive computation of pairwise distances between clusters, the program represents each cluster of sequences as a probabilistic sequence and defines a set of operations to align these probabilistic sequences and to compute the genetic distances between them. CD-HIT and UCLUST utilize a greedy algorithm (Edgar, 2010) that can estimate the similarity between two sequences without performing pairwise alignment of all pairs of sequences. The similarity shared by two sequences is estimated by counting the minimum number of identical short substrings, called 'words', stored in the index table. Only when this number is greater than the required value is an alignment performed to confirm their sequence identity. Using this short word-filtering algorithm (Li *et al.*, 2001, 2002), many unnecessary pairwise alignments can be avoided. In CD-HIT and UCLUST, the sequences are sorted by length so that the longer sequences have a greater chance of becoming representative sequences of clusters, increasing the inclusiveness of clusters. A query sequence is compared to the representative sequence and assigned to the cluster if the similarity between the two sequences is above the predefined threshold; otherwise, the query sequence becomes the representative sequence of a new cluster. These clustering programs have been broadly used in many ecological studies [e.g., MOTHUR (Ling *et al.*, 2010), BLASTClust (Kuenne *et al.*, 2007), CD-HIT (Cameron *et al.*, 2007; Li *et al.*, 2008; Yang *et al.*, 2009), UCLUST (Edgar, 2010)].

In microbial ecology, an OTU often corresponds to a prokaryotic species, which is defined as a group of organisms with high genetic homology. Such a relationship can be defined by either DNA-DNA hybridization (DDH) (Wayne *et al.*, 1987) or 16S rRNA gene sequence similarity because a prokaryotic species is defined as a group of genetically related strains with the type strain as a centroid. In other words, strains of a prokaryotic species are those strains genetically related to the type strain within certain criteria (DDH or 16S rRNA gene similarity values). Therefore, clustering algorithms that utilize the type strain concept in the prokaryotic species definition should be more rational and taxonomically sound.

In this study, we devised a new sequence-clustering algorithm, namely Taxonomy-Based Clustering (TBC), which imitates the prokaryotic classification procedure while omitting the MSA and the calculation of a full distance matrix. The clustering qualities of MOTHUR, BLASTClust, ESPRIT-Tree, CD-HIT and UCLUST were compared with that of the TBC algorithm using test datasets produced by pyrosequencing. The results of our evaluation study showed that TBC provides an accurate estimate of microbial diversity indices with a reasonable computing cost compared with other methods.

Materials and Methods

Sequencing and pre-process of datasets

Three different metagenomic DNAs were extracted from water, soil and kimchi samples using a commercial kit (Mobio). The extracted genomic DNA (gDNA) was amplified using primers targeting the V1 to V3 regions of the bacterial 16S rRNA gene as described previously (Chun *et al.*, 2010). The DNA sequencing was performed using the Roche GS FLX Titanium system according to the manufacturer's instructions. After trimming the barcode, linker, and PCR primer sequences from the raw sequences, the sequences with more than one ambiguous base or read-lengths less than 300 bp were removed from the subsequent analyses. Chimeric sequences were detected by comparison of the identification results of the first and second halves of query sequences. When two identification results indicate taxonomically different taxa (e.g., different orders), the query sequence was considered chimeric and was removed from the final dataset. The resultant sequences were subjected to random subsampling to equalize the sequencing depth to 1,000 reads per sample.

Sequence clustering using conventional methods

Five conventional clustering methods, namely MOTHUR, BLASTClust, ESPRIT-Tree, CD-HIT, and UCLUST, were tested together with TBC for comparative analysis. A 16S rRNA gene sequence similarity value of 0.97 (=97%) was used as the sequence identity threshold for defining species-level OTUs. To produce an input distance matrix for MOTHUR, alignments were performed using ClustalW (version 1.82), and the distance matrix was calculated using DNADIST (version 3.69) in the PHYLIP package (version 3.69) (Retief, 2000). Sequences were assigned to OTUs using the furthest-neighbor clustering algorithm implemented in the MOTHUR package (Schloss *et al.*, 2009). Statistical inferences of species richness including rarefaction analysis (Hurlbert, 1971), Chao1 (Chao, 1984) and ACE (Chao *et al.*, 1993) were performed using MOTHUR.

Taxonomy-Based Clustering (TBC) algorithm

The TBC algorithm is composed of the following steps:

(1) Identical sequences were clustered while ignoring homopolymeric errors.
(2) The longest sequence of each cluster was set as the representative sequence of the given cluster.
(3) The clusters were sorted based on the number of sequences included.
(4) Out of the remaining clusters, the cluster containing the largest number of sequences (query cluster) was searched against representative sequences of all remaining clusters using BLASTN (Altschul *et al.*, 1997). Pairwise nucleotide sequence similarity values were calculated according to Myers and Miller (1988). For a pair of sequences with a certain BLASTN identity value (e.g., 93%), time-consuming pairwise sequence alignment can be omitted.
(5) If two clusters show ≥97% sequence similarity for representative sequences, the cluster with fewer sequences is merged with the larger cluster, and a new database

for the BLASTN search is generated.
(6) Steps (4) and (5) were repeated until the last cluster was considered.

Comparative analysis of clustering results

To assess the quality of the clustering algorithms, a pairwise sequence similarity matrix was created as a reference dataset. The pairwise similarities of all pairwise combinations within a sample was calculated by the global pairwise alignment method (Myers and Miller, 1988). To evaluate the quality of the clustering algorithms resulting in OTU formation, we devised two parameters, namely false conjunction and false disjunction. False conjunction indicates the ratio of the presence of two sequences (=misjoined pair) with <97% sequence similarity in the same OTU and is given by:

$$\sum_{i=1}^{n} (\text{Number of mis - joined pair of sequences in an OTU}_i) \,/\, \sum_{i=1}^{n} (\text{Number of all pair of sequences in an OTU}_i)$$

In contrast, false disjunction indicates the incorrect separation of two sequences into different OTUs when they show ≥97% sequence similarity and is given by:

$$\sum_{i,\, j=1}^{n} (\text{Number of pairs showing pairwise similarities of} \geq 97\% \text{ while each belonging to OTU}_i \text{ and OTU}_j) \,/\, \sum_{i,\, j=1}^{n} (\text{Number of pairwise similarities between sequences in OTU}_i \text{ and OTU}_j)$$

A benchmark study was performed on a Linux CentOS 64 bit server housing four hexa-core Intel Xeon 5300 Series Processors and 64 GB RAM. Raw sequencing datasets from water, soil, and kimchi were used for the performance time evaluation. In the case of MOTHUR, the processing time includes the execution time required for multiple alignments and the generation of the distance matrix.

To compare the clustering outcomes of the different algorithms, a mathematical method, called the OTU profile, was devised to represent the overall OTU profile in a sample (Fig. 1). First, a 1,000 by 1,000 square matrix composed of 1,000 sequences in a sample was constructed for each sample. If sequences n_i and n_j belong to the same cluster, the position (i, j) in the matrix was 1; otherwise, this value was 0. Once the matrices representing the OTU profile were completed, the similarity between these profiles was calculated by dividing the number of the positions (i, j) shared by the two matrices by the total number of sequences in the sample. Pairwise similarity values among the six different algorithms were generated and clustered using the unweighted pair group method with the arithmetic mean (UPGMA).

Implementation and availability

We implemented the method described herein in a software tool. The program was written in JAVA and tested on the Linux and Microsoft Window operating systems. The software and the datasets used in this study are available from the website http://sw.ezbiocloud.net/tbc.

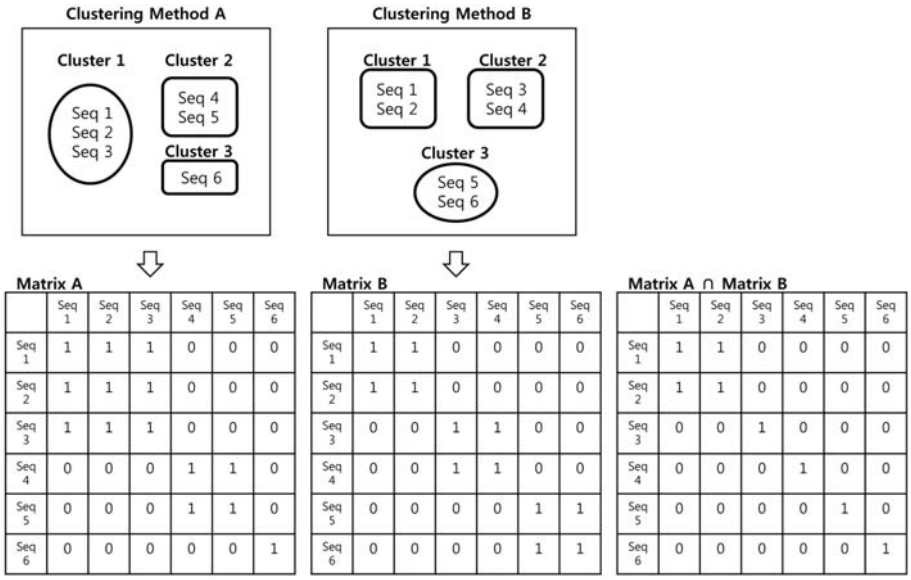
Results and Discussion

Dataset and pairwise sequence similarity matrix for each sample

A number of raw reads were obtained from water (6,339 reads), soil (4,651), and kimchi (2,661) samples, and 1,000 high-quality sequences were randomly subsampled. A reference data matrix comprised 499,500 pairwise similarities within a sample. The percentages of pairwise similarities of ≥97% within each matrix were 6.8, 9.1 and 83.8%, respectively.

Clustering accuracy

The six different clustering algorithms were applied to the subsampled datasets, and the resultant OTUs were compared. The numbers of OTUs produced by different methods are



Matrix A	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6
Seq 1	1	1	1	0	0	0
Seq 2	1	1	1	0	0	0
Seq 3	1	1	1	0	0	0
Seq 4	0	0	0	1	1	0
Seq 5	0	0	0	1	1	0
Seq 6	0	0	0	0	0	1

Matrix B	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6
Seq 1	1	1	0	0	0	0
Seq 2	1	1	0	0	0	0
Seq 3	0	0	1	1	0	0
Seq 4	0	0	1	1	0	0
Seq 5	0	0	0	0	1	1
Seq 6	0	0	0	0	1	1

Matrix A ∩ Matrix B	Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6
Seq 1	1	1	0	0	0	0
Seq 2	1	1	0	0	0	0
Seq 3	0	0	1	0	0	0
Seq 4	0	0	0	1	0	0
Seq 5	0	0	0	0	1	0
Seq 6	0	0	0	0	0	1

Fig. 1. A mathematical way to represent the overall OTU profile to compare the clustering results of different algorithms. An n by n square matrix composed of n sequences in a given sample was constructed for each method. If sequences n_i and n_j belonged to the same cluster (=OTU), the position (i, j) in the matrix was coded 1; otherwise, this value was coded 0. Once the matrices representing the OTU profiles were completed, the similarity between profiles was calculated by dividing the number of the positions (i, j) shared by the two matrices by the total number of sequences in the sample.

Table 1. Numbers of OTUs produced by six different clustering methods. Each dataset contains 1,000 pyro-sequences, and the cutoff for the OTU definition was 0.97 (97%).

Methods	Data sets		
	Water	Soil	Kimchi
TBC	378	265	13
CD-HIT	405	279	20
UCLUST	413	285	19
MOTHUR	382	272	13
BLASTClust	375	282	6
ESPRIT-Tree	371	260	16

summarized in Table 1. The water sample contained the highest number of species, ranging from 375 to 413 OTUs depending on the clustering algorithm used, and the kimchi sample had the lowest values, ranging from 6 to 20 OTUs. In all samples examined in this study, the CD-HIT and UCLUST programs produced the highest number of OTUs. This trend is confirmed by the rarefaction curves (Supplementary data Fig. S1).

The false conjunction ratio values, which indicate the degree of incorrect assignment of two distant sequences in the same OTU, were highest for BLASTClust (12.9% on average), followed by MOTHUR (2.8%), ESPRIT-Tree (2.3%), TBC (1.2%), CD-HIT (0.7%), and UCLUST (0.3%) (Table 2). The poor performance of the BLASTClust program may result from its clustering algorithm, in which a new sequence is joined to a cluster if any member of the cluster has sequence similarity over the cutoff value.

The lowest false disjunction ratio values, which indicate the degree of incorrect separation of similar sequences into different OTUs, was obtained for TBC (3.3% on average), followed by the values for BLASTClust (4.3%) and MOTHUR (6.0%), ESPRIT-Tree (9.72%), UCLUST (43.9%), and CD-HIT (44.7%) (Table 2). The significantly higher values obtained for CD-HIT and UCLUST indicated that the two algorithms produced excessively divided OTUs, many of which should be merged.

Table 2. The accuracy of the six clustering algorithms. False conjunction indicates the ratio of incorrect grouping of distant (<97% similarity) sequences together in the same OTU. False disjunction indicates the ratio of incorrect separation of similar (97% similarity) sequences into different OTUs.

Data sets	Methods	False conjunction (%)	False disjunction (%)
Water	TBC	2.12	7.47
	CD-HIT	0.36	34.47
	UCLUST	0.48	64.14
	MOTHUR	3.48	11.58
	BLASTClust	26.33	8.62
	ESPRIT-Tree	3.92	12.52
Soil	TBC	0.52	1.07
	CD-HIT	1.38	33.87
	UCLUST	0.49	55.66
	MOTHUR	3.74	3.43
	BLASTClust	2.54	1.71
	ESPRIT-Tree	4.92	6.87
Kimchi	TBC	0.94	1.31
	CD-HIT	0.46	65.78
	UCLUST	0.04	11.99
	MOTHUR	1.12	2.94
	BLASTClust	9.92	2.51
	ESPRIT-Tree	1.34	4.79

Benchmarking of the running times

Based on the benchmark study using the subsampled dataset, UCLUST and CD-HIT had the fastest running times. MOTHUR and BLASTClust were extremely slow as a result of the multiple alignments produced and distance matrix generation in MOTHUR and the exhaustive nature of the pairwise sequence comparison in BLASTClust. The TBC method was approximately 1/90-fold slower than UCLUST and approximately 4-, 45-, and 59-fold faster than ESPRIT-Tree, MOTHUR and BLASTClust, respectively.

Similarity of the clustering results among the six different algorithms

To visualize the resemblance of the clustering outcomes of the six clustering algorithms, a UPGMA dendrogram (Fig. 2) was constructed from the similarity values based on the OTU profile method (Fig. 1). Based on the outcome of the clustering, TBC, MOTHUR, and ESPRIT-Tree always produced similar clustering results, whereas CD-HIT and UCLUST generated significantly different sets of OTUs from the same datasets. The latter two programs yielded similar clustering patterns, indicating that they use similar word-based fast clustering algorithms. The clustering outcome of BLASTClust is substantially different from those of all other methods.

Conclusion

In this study, we developed a novel sequence clustering algorithm that mimics the prokaryotic classification principle in which only the genetic similarity between the type strain (=representative sequence) and the query sequence is used for species recognition (OTU formation). Based on the false conjunction and false disjunction ratios, our TBC method showed good performance compared with other clustering methods. The UCLUST and CD-HIT programs are best with respect to computing time but always produce the largest number of OTUs in which two similar (97% similarity) sequences may be assigned to two different OTUs. Based on a comprehensive benchmark study, BLASTClust exhibited the worst performance, with both the slowest run-time and the highest false conjunction ratio.

The generation of OTUs from massive pyrosequencing data is a key step in microbial ecological studies, as important statistical measures are derived from the generated OTU. However, there is no formal criterion specifying the best method by which to cluster sequences into OTUs. Here, we develop and introduce an algorithm that mimics the way we classify bacteria in nature. This classification is possible by (i) assigning a representative sequence for each cluster and (ii) only considering pairwise sequence similarities between these representative sequences and other sequences. Based on a comprehensive comparison with five of the most

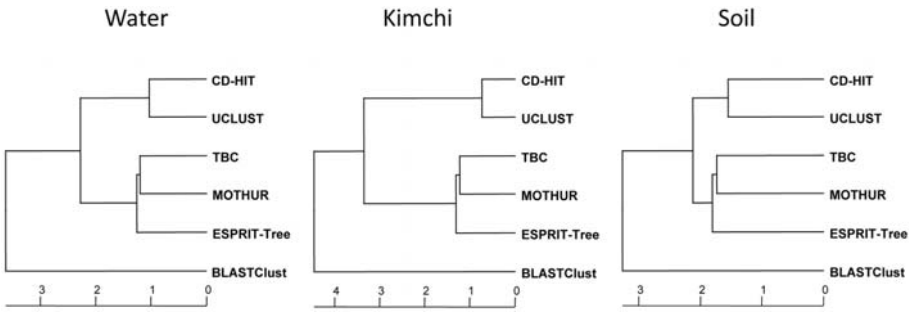


Fig. 2. UPGMA dendrogram showing the resemblance of the clustering results of the six algorithms.

popular clustering programs, it is fair to say that our TBC method generates good-quality clustering, has a reasonable run time and yields outcomes similar to those of MOTHUR (with CLUSTAL MSA), which is the most robust approach, while requiring significantly less computing cost than MOTHUR. The TBC algorithm is implemented in JAVA and can be used as a standalone program in the Linux and Microsoft Windows operating systems.

Acknowledgements

This work was supported by Priority Research Centers Program (#2010-0094020) and a National Research Foundation grant (#2011-0016498) through the National Research Foundation of Korea, funded by the Ministry of Education, Science, and Technology, Republic of Korea.

References

Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D.J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402.

Bacon, D.J. and Anderson, W.F. 1986. Multiple sequence alignment. *J. Mol. Biol.* **191**, 153–161.

Cai, Y. and Sun, Y. 2011. ESPRIT-Tree: hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time. *Nucleic Acids Res.* doi:10.1093/nar/gkr349.

Cameron, M., Bernstein, Y., and Williams, H.E. 2007. Clustered sequence representation for fast homology search. *J. Comput. Biol.* **14**, 594–614.

Chao, A. 1984. Non-parametric estimation of the number of classes in a population. *Scand. J. Stat.* **11**, 265–270.

Chao, A.L. and Lee, S.M. 1992. Estimating the number of classes via sample coverage. *J. Am. Stat. Assoc.* **87**, 210–217.

Chao, A.M., Ma, M.C., and Yang, M.C.K. 1993. Stopping rules and estimation for recapture debugging with unequal failure rates. *Biometrika* **80**, 193–201.

Chun, J., Kim, K.Y., Lee, J.H., and Choi, Y. 2010. The analysis of oral microbial communities of wild-type and toll-like receptor 2-deficient mice using a 454 GS FLX Titanium pyrosequencer. *BMC Microbiol.* **10**, 101.

Edgar, R.C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32**, 1792–1797.

Edgar, R.C. 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* **26**, 2460–2461.

Hamady, M. and Knight, R. 2009. Microbial community profiling for human microbiome projects: Tools, techniques, and challenges. *Genome Res.* **19**, 1141–1152.

Hurlbert, S.H. 1971. The non-concept of species diversity: a critique and alternative parameters. *Ecology* **52**, 577–586.

Kuenne, C.T., Ghai, R., Chakraborty, T., and Hain, T. 2007. GECO – linear visualization for comparative genomics. *Bioinformatics* **23**, 125–126.

Li, W. and Godzik, A. 2006. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**, 1658–1659.

Li, W., Jaroszewski, L., and Godzik, A. 2001. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics* **17**, 282–283.

Li, W., Jaroszewski, L., and Godzik, A. 2002. Sequence clustering strategies improve remote homology recognitions while reducing search times. *Protein Eng.* **15**, 643–649.

Li, W., Wooley, J.C., and Godzik, A. 2008. Probing metagenomics by rapid cluster analysis of very large datasets. *PLoS One* **3**, e3375.

Ling, Z., Kong, J., Liu, F., Zhu, H., Chen, X., Wang, Y., Li, L., Nelson, K.E., Xia, Y., and Xiang, C. 2010. Molecular analysis of the diversity of vaginal microbiota associated with bacterial vaginosis. *BMC Genomics* **11**, 488.

Metzker, M.L. 2010. Sequencing technologies - the next generation. *Nat. Rev. Genet.* **11**, 31–46.

Myers, E.W. and Miller, W. 1988. Optimal alignments in linear space. *Comput. Appl. Biosci.* **4**, 11–17.

Petrosino, J.F., Highlander, S., Luna, R.A., Gibbs, R.A., and Versalovic, J. 2009. Metagenomic pyrosequencing and microbial identification. *Clin. Chem.* **55**, 856–866.

Retief, J.D. 2000. Phylogenetic analysis using PHYLIP. *Methods Mol. Biol.* **132**, 243–258.

Schloss, P.D., Westcott, S.L., Ryabin, T., Hall, J.R., Hartmann, M., Hollister, E.B., Lesniewski, R.A., Oakley, B.B., Parks, D.H., Robinson, C.J., and *et al.* 2009. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.* **75**, 7537–7541.

Thompson, J.D., Higgins, D.G., and Gibson, T.J. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**, 4673–4680.

Wayne, L.G., Brenner, D.J., Colwell, R.R., Grimont, P.A.D., Kandler, O., Krichevsky, M.I., Moore, L.H., Moore, W.E.C., Murray, R.G.E., Stackebrandt, E., and *et al.* 1987. Report of the *ad hoc* committee on reconciliation of approaches to bacterial systematics. *Int. J. Syst. Bacteriol.* **37**, 463–464.

Yang, F., Zhu, Q., Tang, D., and Zhao, M. 2009. Using affinity propagation combined post-processing to cluster protein sequences. *Protein Pept. Lett.* **17**, 681–689.