Published in final edited form as:

J Microbiol Methods. 2013 September ; 94(3): 347–355. doi:10.1016/j.mimet.2013.07.004.

MSClust: A Multi-Seeds Based Clustering Algorithm for microbiome profiling using 16S rRNA Sequence

Wei Chen^{1,2}, Yongmei Cheng¹, Clarence Zhang³, Shaowu Zhang¹, and Hongyu Zhao^{2,*}

¹College of Automation, Northwestern Polytechnical University, 710072, Xi'an, China

²Department of Biostatistics, Yale School of Public Health, New Haven, CT 06510, US

³Keck Biotechnology Laboratory, Biostatistics Resource, Yale School of Medicine, New Haven, CT 06510

Abstract

Recent developments of next generation sequencing technologies have led to rapid accumulation of 16s rRNA sequences for microbiome profiling. One key step in data processing is to cluster short sequences into operational taxonomic units (OTUs). Although many methods have been proposed for OTU inferences, a major challenge is the balance between inference accuracy and computational efficiency, where inference accuracy is often sacrificed to accommodate the need to analyze large numbers of sequences. Inspired by the hierarchical clustering method and a modified greedy network clustering algorithm, we propose a novel multi-seeds based heuristic clustering method, named MScLust, for OTU inference. MScLust first adaptively selects multi-seeds instead of one seed for each candidate cluster, and the reads are then processed using a greedy clustering strategy. Through many numerical examples, we demonstrate that MScLust enjoys less memory usage, and better biological accuracy compared to existing heuristic clustering methods while preserving efficiency and scalability.

Keywords

Clustering Algorithms; Operational Taxonomy Unit (OTU); Next-generation Sequencing; Seeds-Selection; 16S rRNA Reads

1. Introduction

Microbiome research has attracted much attention in recent years largely due to the rapid developments of next-generation sequencing technologies through which vast amounts of DNA and RNA sequences have been collected. Although these data provide rich information on microbial composition, abundance, and effects on human health, the analyses of these data present many computational and statistical challenges. A key step in these

© 2013 Elsevier B.V. All rights reserved.

*Corresponding author: hongyu.zhao@yale.edu.

Competing interests We declare that we have no competing interests.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Availability and implementation: The program is written in Matlab and we are still optimizing the algorithm. The beta version of the algorithm is available at <http://bioinformatics.med.yale.edu/group/>

analyses is the assignments of sequences to different species and/or the clustering of sequences. An appropriate clustering method is needed to characterize a microbial community, such as its richness, diversity and structure [1]. In this context, the operational taxonomic units (OTUs) inferred from sequence clusters are often used as separate entities in downstream analysis, and as such OTU estimation is a critical step in microbiome analysis.

Two complementary approaches are commonly used to characterize the microbial communities for 16S rRNA sequences: taxonomy-dependent methods and OTU-based methods. For taxonomy-dependent methods, a query sequence is mapped to the best-matched sequence in the reference database. Thus, the usefulness of this class of methods depends on the completeness of the reference database. For OTU-based methods, they can be categorized into three classes: hierarchical clustering, heuristic clustering and model-based methods. Hierarchical clustering methods require a full distance matrix (e.g. Mothur [2]), which is derived either from pair-wise alignment or multiple alignments [3, 4]. Thus, it has a quadratic computational complexity both in time and space, $O(N^2)$, where N is the number of sequences. Even with improved strategies such as adopting a k-mer distance to filter out unnecessary pair-wise alignment and/or to store the reduced-distance with a sparse matrix (e.g. ESPIRIT [5]), the computational complexity remains $O(N^2)$. In order to address the time and memory bottleneck in hierarchical clustering algorithms, heuristic clustering methods were developed using a greedy clustering strategy (e.g. CDHIT [6] and Uclust [7]). These methods first select an input sequence as a seed for the initial cluster, and each input sequence is then examined sequentially. An input sequence is added to one existing cluster if the distance between the query sequence and representative sequences of existing clusters is within a pre-defined threshold, otherwise a new cluster is created and the query sequence becomes the new representative sequence. As a result, heuristic clustering methods have a lower complexity of $O(LN)$ (where N is the number of sequences and L is the read length) at the cost of accuracy. Because the Needleman-Wunsch algorithm [7] used for sequence alignment in both hierarchical clustering and heuristic clustering methods is the rate limiting step, some algorithms have been introduced to reduce the time-consuming alignment process [8, 9]. For example, GramCluster uses a grammar-based distance rather than alignment-based distance metric to cluster sequences. The last OTU-based category clustering is model-based clustering methods which were developed to address the need to select a constant and optimal threshold to define OTUs. For example, CROP [10] represents the sequence by a Gaussian mixture model. Compared to hierarchical and heuristic clustering methods, CROP adopts a soft-threshold to clustering sequences. It can achieve a better cluster quality when the data sets have low complexity but have worse performance for high complexity datasets [11].

Because most heuristic clustering methods select only one seed for each cluster, the results are sensitive to the selected seeds that represent the clusters. For example, a change in the order of the input sequences may alter the clustering results significantly. Hierarchical clustering methods (either based on average-linkage or complete-linkage) consider all the sequences in a cluster when forming clusters. Inspired by SPICI – a local network clustering method [12], which generates clusters from local density community based on a greedy clustering method, we propose a **Multi-Seeds based Clustering** algorithm, MSClust, which incorporates an adaptive seed-selection procedure and a greedy heuristic clustering procedure. Through extensive simulations, we demonstrate that our method can achieve a balance between cluster quality and time complexity as compared to traditional heuristic clustering methods. Furthermore, its performance is similar or equivalent to hierarchical clustering-based methods but it requires much less memory usage. Overall, MSClust is a good alternative to existing OTU-based clustering methods. The rest of the paper is organized as follows: In the “Algorithm” section, we outline the proposed approaches. In the

“Results and Discussions” section, we compare our method with other methods based on different experiments.

2. Algorithm

The main motivation of our method is to overcome the sensitivity of seed selection in traditional heuristic clustering methods, as well as the large memory usage of hierarchical clustering methods. In this section, we describe MSClust, an adaptive multi-seeds based heuristic clustering method that avoids the large memory need for storing seeds and/or distance matrix. The MSClust algorithm has three parameters: *Sim*, the similarity threshold for inferring OTUs; *Tsize*, the maximum size of Temp Cluster which is used to ensure that the Temp Cluster contains enough reads to yield a reliable candidate seed set (default value 10); and *NumS*, the number of selected seeds to represent each candidate cluster (the default value is 3). These settings have robust performance as demonstrated through the following experiments compared to species-level ground-truth. The flowchart for the proposed method is shown in Figure 1.

2.1 Algorithm overview

MSClust uses a greedy heuristic strategy to build one cluster at a time. Each cluster is expanded from a limited initial set with multi-seeds, where the initial multi-seeds are generated based on an adaptive strategy. Unassigned sequences are then compared to the seeds sequentially. A new sequence is added to the current cluster and removed from the input if the average distance between the sequence and seeds is smaller than the user-defined threshold; otherwise, the sequence is marked as unassigned.

2.2 Input and Initial & Termination check

In practice, in order for the more abundant sequences to be preferentially selected, and to reduce the time complexity, we first sort the input sequences in descending order of their abundance, and then return the unique sequences of the sorted sequences. Once a cluster is completed, the next cluster will be generated based on the remaining unassigned sequences. The process is repeated until all the sequences are clustered.

2.3 Seeds Selection

Traditional heuristic clustering-based methods randomly select a sequence as a seed, thus the clustering results are sensitive to the seed choice. In contrast, MSClust uses an adaptive strategy for seeds-selection. First, we randomly select a sequence as a Temp Seed, then sequences are assigned to the Temp Seed if the distance is within a pre-defined threshold to generate the Temp Cluster. When the size of the Temp Cluster reaches a pre-defined threshold size, the adaptive seeds-selection procedure is triggered to generate Candidate Seeds. A core seed is determined as the assigned sequences that minimize the average distance with the others in the Temp Cluster. Then from the Temp Cluster, we repeatedly choose one of the *NumS* nearest sequences from the ones selected so far (*NumS* is parameter to define a maximum seed for each cluster). When *NumS* = 1, our algorithm can be regarded as a modified heuristic clustering-based method, such as Uclust and CDHIT. Thus, MSClust can be thought as a more general approach. The pseudo-code for the seeds selection procedure is shown in Figure 2.

2.4 Cluster Formation

MSClust is primarily a greedy clustering algorithm. The main difference between the proposed and traditional heuristic methods is that one cluster is constructed based on multi-seeds instead of one seed each time. When the Candidate Seeds for each cluster are

determined, the cluster grows gradually by adding sequences one by one based on the similarity threshold Sim . At each step, a new input sequence is compared to the Candidate Seeds. If the average distance $D = \text{average}(s, (\text{Candidate Seeds}))$ is smaller than the pre-defined threshold, the sequence is added to the cluster. Otherwise, the checked sequence is marked as unassigned. This procedure is repeated until all input sequences are checked. Once it is done, the unassigned sequences are indexed as input sequences. Meanwhile, in the early step of constructing Temp Cluster, we incorporate the k-mer distance to identify promising seed/read pair and filter large part of unnecessary pair-wise alignment. If the k-mer distance between the seed and sequence is smaller than d (default value is 0.5), then the Needleman-Wunsch algorithm is used to compute the dissimilarity.

3. Results and Discussions

To assess the performance of MSClust, we conducted experiments to compare MSClust with seven commonly used OTU inference methods: Mothur [2], CDHIT [6], Uclust [7], GramCluster [8], DNAClust [9], ESPRIT [5] and ESPRIT-Tree [13] based on a number of benchmark datasets, including a generated dataset from published sequences, a simulated dataset generated from 454sim software [14] and two published datasets, Stacked_60 [10] and Clone43 [15]. CDHIT, Uclust, GramCluster and DNAClust are heuristic clustering methods commonly used in analyzing 16S rRNA reads. Mothur and ESPRIT are two commonly used hierarchical clustering methods while ESPIT-Tree is a modified version of ESPRIT. To make relatively fair comparisons across different methods, we used the default parameter settings for all the OTU-based algorithms in our analysis.

To evaluate clustering performance, NID score and NMI score [16, 17] were used to assess clustering quality. Assume that there are N reads from m species ($S_1, S_2, \dots, S_m$) and they are clustered into n clusters ($C_1, C_2, \dots, C_n$) at a specific dissimilarity threshold by a given clustering algorithm. Let $|S_i|$ denote the true number of reads from species i , $|C_j|$ denote the number of reads from cluster j , and a_{ij} denote the number of reads from species i and categorized into cluster j . Then the NID and NMI are formally defined as follows:

$$I(S, C) = \sum_{i=1}^m \sum_{j=1}^n \frac{a_{ij}}{N} \log \frac{a_{ij}/N}{|S_i| |C_j| / N^2}, \quad i=1, 2, \dots, m; j=1, 2, \dots, n$$

$$H(S) = - \sum_{i=1}^m \frac{S_i}{N} \log \frac{S_i}{N}, \quad H(C) = - \sum_{j=1}^n \frac{C_j}{N} \log \frac{C_j}{N}$$

$$NID = 1 - \frac{I(S, C)}{\max(H(S), H(C))}$$

$$NMI = \frac{2I(S, C)}{H(S) + H(C)}$$

NMI measures the degree of information provided by the clustering results with respect to the ground truth where a larger value implies better clustering results. NID score, an information theoretic-based measurement which stratifies both normalization and metric properties, is regarded as the best measurement for cluster validation [16], where a smaller NID value indicates better clustering quality. They both have a value range [0, 1].

Experiment 1: V6 Variable region of 16S rRNA from Turnbaugh et al.'s Gut dataset

The first benchmark dataset (V6) (~460K) was constructed based on the human gut microbiome dataset [18] by running MegaBlast against the RDP Database [19] to retain the annotated sequences if the similarity is greater than 97%. Additionally, sequences that are annotated with more than one taxonomy label at the species level were removed. This resulted in a total of 310K reads, classified into 177 species. The average sequence length was 121nt. To obtain robust results, each method was run 10 times and 30K reads were randomly sampled from the annotated V6 benchmark database in each run.

Figure 3 depicts the average number of inferred OTUs as a function of similarity cut-off over 10 runs. The estimated numbers of OTUs were significantly different at high similarity levels which usually correspond to genus and/or species. These results suggest that we should be cautious in estimating microbiome community diversity at lower taxon level. For lower similarity cut-offs, the results of MSCLust were similar to those of ESPRIT-Tree. At the same cluster level, MSCLust inferred a smaller number of OTUs than other algorithms except GramCluster in the similarity interval [96%, 98%]. Although it still tended to overestimate the number of OTUs, MSCLust was less biased. For similarity levels ranging from 96% to 98%, GramCluster inferred a smaller number of OTUs, partly due to the fact that the similarity in GramCluster is generated from grammar-distances, which are based on Lempel-Ziv complexity [20]. It tended to be smaller than that computed through a pairwise sequence alignment [8] used in the proposed algorithm. In Figure 3, there is no curve for Mothur because it failed to cluster the reads into OTUs, even with the unique reads as input. Therefore, Mothur was not suited for processing large scale 16S rRNA datasets. In microbiome analysis, 97% is often used as the similarity cutoff to define OTUs at species level, which is the basis for estimating the diversity and composition of microbial community, though the definition is controversial [17, 21, 22]. Table 1 summarizes the number of OTUs inferred at the 97% similarity level. We observe that the estimated numbers of OTUs were larger than those expected (132) at 97% similarity cut-off and varied significantly across different algorithms. The maximum number was ~3,000 estimated by Uclust whereas the minimum was ~1,100 inferred by GramCluster.

The estimated numbers of OTUs offer limited information of the quality of the clustering methods [17], thus we further calculated the NMI and NID scores to evaluate the clustering performance (Figure 4). Since a lower cut-off (< 90%) sometimes corresponds to family or even higher taxa levels, using the species ground-truth to compute NID and NMI scores is not proper, thus we did not use such values. We can see that the NID curve of MSCLust is close to that of ESPRIT-Tree while below others. The minimum NID score, which by definition corresponds to the best clustering results for the compared algorithms, was achieved at different cut-off values for different algorithms. It suggests that there is no consistent similarity cut-off for all the algorithms, partly due to the different strategies for measuring similarity and clustering methods incorporated in each algorithm. As expected, the minimum value for MSCLust was 0.05 at 93% similarity, which was a bit greater than that of ESPRIT-Tree (0.035), while the minimum values for the other algorithms were greater than 0.1 (Figure 4(a)). This suggests that the proposed method has a better cluster performance than CDHIT, Uclust, GramCluster, DNAClust and ESPRIT. We found that MSCLust had higher NMI scores, and the NMI curve is similar to ESPRIT-Tree. It suggests that MSCLust outperforms other clustering algorithms, including the heuristic clustering algorithms and ESPRIT. In addition, the maximum NMI scores vary with the algorithms. For example, the largest NMI score was 0.97 inferred by ESPRIT-Tree at 93% similarity cut-off, followed by 0.95 inferred by MSCLust at 93% similarity cut-off while others returned smaller NMI scores at lower similarity cut-offs (Figure 4(b)), which also suggests that MSCLust performed better than GramCluster, DNAClust, CDHIT, Uclust and ESPRIT. These results show that MSCLust can achieve a higher accuracy for analyzing large-scale microbiome data. Meanwhile, an interested finding is that the inferred numbers of OTUs at those similarities where each algorithm had the maximum NMI score were much closer to the expected number (Table 1), especially for MSCLust (140), ESPRIT-Tree (141) and CDHIT (126). In the context of inferring the numbers of OTUs among the algorithms compared, similar trends were found as those based on NMI score at those similarities where the minimum NID score were achieved, except that ESPRIT, Uclust and GramCluster returned different estimated numbers of OTUs. These results suggest that the similarity at which the minimum NID score or the maximum NMI score was achieved is a better choice for inferring OTUs at species level rather than the commonly used 97% similarity cut-off.

Table 2 shows the time cost results for different algorithms at those similarity thresholds where the minimum NID scores were achieved. We can see that heuristic clustering methods took less time to cluster sequences (< 1 minute) than MSCLust (0.26h), which gave smaller NID score. The time cost of MSCLust was much smaller than both ESPRIT (2.9h) and half of that of ESPRIT-Tree (0.45h).

Besides the similarity cut-off for defining OTUs, there are two parameters in our algorithm, *NumS* and *Tsize*. In order to check how the two parameters affect the cluster results, we first tested the effect of the number of seed (*NumS*) on the NID and NMI scores with a fixed *Tsize* (Figure 5). We found only slight changes on NID score when different *NumS* values were used. With the increase of *NumS* from 3 to 9, the NID score became smaller (~4%), which suggests that higher *NumS* may improve cluster performance. When the *NumS* increased from 3 to 9, the NMI score became larger (~1%). We also examined the effect of *Tsize*. When we altered the *Tsize* from 5 to 20, the NMI score and NID score were almost the same (0.5% and 1%) (Figure 6).

Experiment 2: Simulated dataset

In the above study, the ground truth was established based on running a blast to the reference database, which may not be all correct. In this section, we use simulated dataset to evaluate the clustering results where the ground truths were directly taken from simulated data. The simulated dataset was generated by 454sim, a software that simulates data using configurable statistical models to accommodate different numbers of sequence reads, sequence lengths, errors rates, and abundance. The phylogenetic tree for the initial species sequences is shown in Figure 7, where the minimum distance among different species sequences is 0.02 and the maximum distance is 0.23. The details of the simulated datasets are summarized in Table 3.

As shown in Figure 8, the relative performance of different methods was similar to that observed from the first experiment. Both NID and NMI curves show that MSCLust had a similar performance with ESPRIT-Tree and Mothur. The minimum NID score of MSCLust was 0.025, a bit smaller than those of CDHIT (0.03), Uclust (0.043), GramCluster (0.039), DNAClust (0.027), and ESPRIT (0.28). The maximum NMI score of MSCLust was similar to those of ESPRIT-Tree and Mothur (0.996, 0.993 and 0.999). It is a bit greater than other methods (0.964 for Uclust, 0.986 for CDHIT, 0.98 for GramCluster, 0.981 for DNAClust, and 0.8 for ESPRIT). Those results confirm that MSCLust can achieve a higher clustering accuracy than traditional heuristic clustering methods.

Experiment 3: Barriuso's Stacked_60 dataset

We further considered another benchmark data set, namely the Stacked_60 dataset covering the V6 region from Barriuso [11]. The average read length is 98 nt and the expected OTU numbers at the 97%, 95% and 90% similarity thresholds are 59, 57 and 52, respectively. As for the number of OTUs estimated by different methods (Table 4), GramCluster inferred a large number of OTUs at the 97% similarity, 929, which is about 16 times as the expected number, which is 59. When the threshold was set to 95% or 90%, the estimated numbers of OTUs for all methods were closer to the correct numbers. MSCLust inferred 59, 58 and 52 OTUs at the 97%, 95% and 90% similarity thresholds respectively, which was quite on target.

Figure 9 illustrates the NMI and NID scores for different methods. Similar to experiments 1 and 2, both NID and NMI curves show that MSCLust can achieve an equivalent performance as hierarchical clustering methods. The minimum NID score of MSCLust was the same to those of ESPRIT, ESPRIT-Tree and Mothur (0.0056) and a bit smaller than the heuristic

clustering algorithms except GramCluster (0). MSCLust, ESPRIT, ESPRIT-Tree and Mothur had the same maximum NMI score (0.997), indicating similar cluster quality.

Experiment 4: Short length read dataset Clone43

In order to investigate the performance for shorter reads, we considered an additional experiment dataset Clone43. As described by Huse [15], this dataset consists of 202,340 reads from a mixture of 43 plasmid clones spanning the V6 region of 16S rRNA gene, with an average length of 61 nt. Because of the lack of ground-truth, i.e. species origin that each read belongs to is unknown, we used the estimated number of OTUs to evaluate the clustering quality (Table 5 and Figure 10). At the commonly used threshold 97%, the smallest number of OTUs was 708 returned by Mothur, followed by ESPRIT-Tree (1096) and MSCLust (1351). The largest number was returned by ESPRIT (4397). We can see that MSCLust can reduce the OTU inflation, had a similar performance as ESPRIT-Tree, and was better than other heuristic clustering methods.

To evaluate the biological implications of the inferred OTUs, we mapped the representative sequences of each OTU to their nearest known species (Table 6). Among the 1351 OTUs returned by MSCLust as discussed above, 43 OTUs can be mapped to their nearest template, which is similar to the result from Mothur (43/720). That is the OTU results can completely cover all the representative species, and the accuracy is higher than other methods as shown in this table. When a mapping error rate of 3% was allowed, 209 out of 1351 OTU could be mapped to their templates, indicating that some reads from the same species are sub-clustered into different OTUs. For the other OTUs that could not be mapped, most of them (799/1142) were singletons, containing only one read, which resulted from the accumulation of sequencing errors.

Algorithm Complexity—The time complexity of MSCLust has two components. For Multi-seeds selection, a total of $Tsize$ reads are chosen. In the worst case, all the sequences are scanned, thus the maximum complexity is in the order of $O(N)$, where N is the total number of unique sequences, often much smaller than the total number of input sequences M , i.e. $N \ll M$. Next, in order to find the $NumS$ Candidate Seeds, a $Tsize \times Tsize$ matrix is generated and $NumS$ sequences are chosen, so the worst-case complexity is $O(N(Tsize^2 + NumS \log(Tsize)))$. Finally, suppose the average number of Clusters is K and each sequence is compared with the Candidate Seeds, resulting in a complexity $O(K \cdot NumS \cdot N)$. In summary, the total complexity for MSCLust algorithm is $O(N(Tsize^2 + NumS \log(Tsize))) + K \cdot NumS \cdot N$, because $Tsize$ and $NumS$ are smaller and $K \ll N$, which the most significant component has a time complexity of the order of N . This shows that MSCLust has a little larger complexity than traditional heuristic clustering methods CDHIT and Uclust, but a smaller complexity than hierarchical clustering methods such as Mothur and ESPRIT ($O(N^2)$).

Memory usage is a key computation issue which should be carefully considered for processing sequencing data because sometimes we should load the distance matrix into memory before processing clustering resulting in a large memory usage. For example, Mothur needs 100G when processing a full run 454 data. Suppose the average sequence length is L , the total clustering number is C and each sequence is stored in $O(L)$. Then for MSCLust, in each cycle, we only store the limited Candidate Seeds ($NumS$) resulting in a memory complexity of $O(NumS \cdot L)$, which suggests that MSCLust has even a smaller memory requirement than traditional greedy heuristic clustering methods ($O(LC)$), especially for a large scale data set where $NumS \ll C$. The memory usage for MSCLust tends to be a constant value which suggests that there is no memory bottleneck for MSCLust method.

4. Conclusions

With rapid accumulation of 16S rRNA sequences and whole genome shotgun sequences, there is a great need to develop effective clustering methods to analyze these data. In the present work, we have proposed a novel clustering method which enables us to cluster millions of 16S rRNA sequences with high accuracy, low memory usage and computational complexity. Different from traditional heuristic clustering algorithms, our method integrated an adaptive seed-selection strategy and used a modified greedy heuristic clustering procedure. In this algorithm, multiple seeds were selected for each cluster, rather than one single seed. Together with the fact that an analogue “average-linkage” clustering was used in processing of heuristic clustering, our method can achieve higher-quality clusters. Meanwhile, since one cluster is generated in each iteration, it overcomes the bottleneck of space complexity and it has a very small memory usage. Therefore, our method is more accurate and it could be efficiently used to handle large-scale data sets.

Acknowledgments

We would like to thank the Yale University Biomedical High Performance Computer Center for computation support. This work was supported in part by NIH GM59507, UL1 RR024139, National Natural Science Foundation of China (No. 61170134), Aeronautical Science Foundation of China (No.20100853010) and Doctorate Foundation of Northwestern Polytechnical University (No. cx201017).

References

1. Sharpton TJ, Riesenfeld SJ, Kembel SW, Ladau J, O'Dwyer JP, Green JL, Eisen JA, Pollard KS. PhylOTU: A High-throughput procedure quantifies microbial community diversity and resolves novel taxa from metagenomic data. *PLOS computational Biology*. 2011; 7(1):e1001061. [PubMed: 21283775]
2. Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ, et al. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl Environ Microbial*. 2009; 75(23):7537–7541.
3. Schloss PD, Handelsman J. Introducing DOTUR, a computer program for defining operational taxonomic units and estimating species richness. *Appl Environ Microbial*. 2005; 71(3):1501–1506.
4. Huse SM, Welch DM, Morrison HG, Sogin ML. Ironing out the wrinkles in the rare biosphere through improved OTU clustering. *Environ Microbial*. 2010; 12(7):1889–1898.
5. Sun Y, Cai Y, Liu L, Yu F, Farrell ML, McKendree W, Farmerie W. ESPRIT: estimating species richness using large collections of 16S rRNA pyrosequences. *Nucleic Acids Res*. 2009; 37(10):e76. [PubMed: 19417062]
6. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*. 2006; 22(13):1658–1659. [PubMed: 16731699]
7. Edgar RC. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*. 2010; 26(19):2460–2461. [PubMed: 20709691]
8. Russell DJ, Way SF, Benson AK, Sayood K. A grammar-based distance metric enables fast and accurate clustering of large sets of 16S sequences. *BMC Bioinformatics*. 2010; 11:601. [PubMed: 21167044]
9. Ghodsi M, Liu B, Pop M. DNACLUSt: accurate and efficient clustering of phylogenetic marker genes. *BMC Bioinformatics*. 2011; 12:271. [PubMed: 21718538]
10. Hao X, Jiang R, Chen T. Clustering 16S rRNA for OTU prediction: a method of unsupervised Bayesian clustering. *Bioinformatics*. 2011; 27(5):611–618. [PubMed: 21233169]
11. Barriuso J, Valverde JR, Mellado RP. Estimation of bacterial diversity using next generation sequencing of 16S rDNA: a comparison of different workflows. *BMC Bioinformatics*. 2011; 12:473. [PubMed: 21219653]

12. Peng J, Mona S. SPICi: a fast clustering algorithm for large biological networks. *Bioinformatics*. 2010; 26(8):1105–11. [PubMed: 20185405]
13. Cai Y, Sun Y. ESPRIT-Tree: hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time. *Nucleic Acids Res*. 2011; 39(14):e95. [PubMed: 21596775]
14. Lysholm F, Andersson B, Persson B. An efficient simulator of 454 data using configurable statistical models. *BMC research notes*. 2011; 4(1):449. [PubMed: 22029428]
15. Huse SM, Huber JA, Morrison HG, Sogin ML, Welch DM. Accuracy and quality of massively parallel DNA pyrosequencing. *Genome Biol*. 2007; 8(7):R143. [PubMed: 17659080]
16. Vinh, Nguyen Xuan; Epps, Julien. Information Theoretic Measurement for clustering comparison: Variants, Properties, Normalization and Correction Chance. *Journal of Machine Learning Research*. 2010; 11:2837–2854.
17. Sun Y, Cai Y, Huse SM, Knight R, Farmerie WG, Wang X, Mai V. A large-scale benchmark study of existing algorithms for taxonomy-independent microbial community analysis. *Briefings in Bioinformatics*. 2012; 13(1):107–121.
18. Turnbaugh PJ, et al. A core gut microbiome in obese and lean twins. *Nature*. 2009; 457:480–484. [PubMed: 19043404]
19. Cole JR, et al. The Ribosomal Database Project: improved alignments and new tools for rRNA analysis. *Nucleic Acid Res*. 2009; 37:D141–D145. [PubMed: 19004872]
20. Lempel A, Ziv J. On the Complexity of Finite Sequences. *IEEE Transactions on Information Theory*. 1976; 22:75–81.
21. Schloss PD, Westcott SL. Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Appl. Environ. Microbial*. 2011; 77(10): 3219–3226.
22. Sogin ML, Morrison HG, Huber JA, Mark WD, Huse SM, Neal PR, Arrieta JM, Herndl GJ. Microbial diversity in the deep sea and the underexplored “rare biosphere”. *Proc Natl Acad Sci U S A*. 2006; 103(32):12115–12120. [PubMed: 16880384]

Highlights

- A multi-seeds based heuristic clustering method is proposed.
- It incorporates an adaptive seed-selection procedure and a greedy heuristic clustering procedure.
- An analogue “average-linkage” clustering strategy rather than single-linkage was used in processing of heuristic clustering.
- It can achieve a balance between cluster quality and time complexity as compared to traditional heuristic clustering methods, meanwhile, its performance is similar or equivalent to hierarchical clustering-based methods but it requires much less memory usage.

Figure 1.
Flow chart of the MSClust algorithm

Algorithm: Seeds Selection (N , $threshold$, $Tsize$, $NumS$)

Choose $s \in N$ at random

$Temp Cluster = \{s\}$

While $|Temp Cluster| < Tsize$

If $distance(s_i, s) \leq threshold$

$Temp Cluster = Temp Cluster \cup s_i$

End if

End while

$c = \arg_{s_i} \min_{s_i \in Temp Cluster} (average_distance(s_i, Temp Cluster))$

Candidate = {c}

For $i=1: NumS$

$s_i = \arg_{s_i} \min(average_distance(s_i, Candidate))$

Candidate = Candidate $\cup s_i$

End for

Figure 2.

The pseudo-code for the Seeds Selection procedure.

Figure 3. Comparison of the Number of OTUs predicted by ESPRIT, CDHIT, Uclust, GramCluster, MSClust, DNA Clust and ESPRIT-Tree using the V2 benchmark at different sequences similarity thresholds ranging from 0.98 to 0.9

Figure 4. NMI and NID scores for different algorithms based on the species ground-truth (a). NID scores across different algorithms at different thresholds. (b). NMI scores across different algorithms at different thresholds.

Figure 5. The effect of Parameter NumS (Number of Seeds) on clustering results. a). NID scores of MSClust are calculated at various similarity levels with different NumS. b) NMI scores of MSClust are compared for various NumS.

Figure 6.

The effect of Parameter Tsize (size of Temp Cluster) on clustering results. a) NID scores of MSClust are calculated at various similarity levels with different Tsize. b) NMI scores of MSClust for various NumS with the same NumS=3.

Figure 7.
phylogenetic tree for the 30 species

Figure 8. NID score and NMI score for the simulated data set. (a). NID scores for the compared algorithms at different thresholds. (b). NMI scores for the compared algorithms at different thresholds.

Figure 9. NMI and NID Scores for different algorithms based on the benchmark dataset Stacked_60 with species ground-truth. (a) NID Scores across different algorithms at different thresholds. (b) NMI Scores across different algorithms different thresholds.

Figure 10.

Table 1

Number of OTUs inferred at 97% similarity level and the similarity levels where the NIM Score is maximized and the NID score is minimized

	ESPRIT	CDHIT	Uclust	GramCluster	MSClust	DNAClust	ESPRIT-Tree	Mothur
Expected Number of OTUs					132			
97% similarity	2441	2030	2996	1092	1760	2840	1950	NA
Maximum NMI Score	69	126	100	332	140	90	141	NA
Minimum NID Score	83	126	74	171	140	99	141	NA

NA: Fail to cluster the reads into OTUs

Table 2

Time cost for the compared methods at those similarities where achieved the minimum NID scores

Algorithms (parameters)	NID scores	Computational time
¹ Uclust (10%)	0.19	3s
DNAClust (10%)	0.18	5s
CDHIT (10%)	0.11	7s
GramCluster (9%)	0.20	10s
MSClust (93%)	0.05	0.26h
ESPRIT-Tree (92%)	0.035	0.45h
ESPRIT (10%)	0.25	2.9h
Mothur	NA	NA

¹ Similar to otupipe used in Barriuso's work. ESPRIT and ESPRIT-Tree provide all the clustering results from 1% to 15% similarity threshold at one time. NA: Fail to cluster the reads

Table 3

Details on the simulated datasets

Simulated data set	
# of species sequence	30
Species length	227 ~ 243
Speices distance ranges	0.02~0.23
Total sequences	294,548
Ratio distribution ranges	0.027 ~ 0.041

* Ratio I = (# of sequences simulated from species I) / total simulated sequences

Table 4

The OTU results based on Stacked_60 dataset

	93% similarity	95% similarity	90% similarity
Expected number of OTU	59	57	52
Uclust	62	59	54
DNAClust	59	59	56
CDHIT	77	59	55
GramCluster	929	70	59
MSClust	59	58	52
ESPRIT-Tree	60	59	50
ESPRIT ^a	52	52	50
Mothur	62	59	54

^a the results are different Barriuso's work may be due to the modification in ESPRIT software.

Table 5

The OTU results for real dataset Clone43

	ESPRIT	CDHIT	Uclust	GramCluster	MSClust	DNAclust	ESPRIT-Tree	Mothur
Expected Number of OTUs					43			
97% similarity	4397	1464	1883	2071	1351	3658	1096	708
# of Singleton at 97% similarity	1496	744	411	893	835	0	780	413

Table 6

Mapped representative sequence of each cluster to the template sequences

	Clone43	
	# of Exactly Mapped	# of Mapped (3% dissimilarity^a)
Mothur	43/720	87/720
ESPRIT	24/4397	2633/4397
ESPRIT-Tree	12/1096	87/1096
Uclust	1/1883	385/1883
CD-HIT	1/1464	77/1464
DNAClust	0/3658	1079/3658
GramClust	39/2071	1212/2071
MSClust	43/1351	209/1351

The sequence identity is defined as the number of insertions, deletions and substitutions in a sequence as compared with its template but and the terminate gap will not be accounted, divided by the length of the template sequence.