J Biosci (2019) 44:148
DOI: 10.1007/s12038-019-9964-5

A critical analysis of state-of-the-art metagenomics OTU clustering algorithms

Ashaq Hussain Bhat[1], Punithaa Prabhu[1]* and Kalpana Balakrishnan[2]

[1]*K.S. Rangasamy College of Technology, Tiruchengode 637 215, India*

[2]*Indian Institute of Technology Madras, Chennai, India*

**Corresponding author (Email, spunitha156@gmail.com)*

MS received 10 March 2019; accepted 9 August 2019; published online 6 November 2019

Taxonomic profiling, using hyper-variable regions of 16S rRNA, is one of the important goals in metagenomics analysis. Operational taxonomic unit (OTU) clustering algorithms are the important tools to perform taxonomic profiling by grouping 16S rRNA sequence reads into OTU clusters. Presently various OTU clustering algorithms are available within different pipelines, even some pipelines have implemented more than one clustering algorithms, but there is less literature available for the relative performance and features of these algorithms. This makes the choice of using these methods unclear. In this study five current state-of-the-art OTU clustering algorithms (CDHIT, Mothur's Average Neighbour, SUMACLUST, Swarm, and UCLUST) have been comprehensively evaluated on the metagenomics sequencing data. It was found that in all the datasets, Mothur's average neighbour and Swarm created more number of OTU clusters. Based on normalized mutual information (NMI) and normalized information difference (NID), Swarm and Mothur's average neighbour showed better clustering qualities than others. But in terms of time complexity the greedy algorithms (SUMACLUST, CDHIT, and UCLUST) performed well. So there is a trade-off between quality and time, and it is necessary while analysing large size of 16S rRNA gene sequencing data.

Keywords. Bioinformatics pipelines; clustering; metagenomics algorithms; microbiome; next generation sequencing; operational taxonomic units

1. Introduction

Bioinformatics is an assemblage of computer science, mathematics, statistics and biotechnology. Metagenomics is one of sub areas of bioinformatics which deals with the study of microorganism in their respective environments. Microbiome is considered to be the dark matter of biological universe as most of it is still unknown and only a part can be cultured *in vitro* (Bernard *et al.* 2018; Lok 2015; Kellenberger 2001). Microorganisms play a prominent role in every part of biosphere, in creating different types of medicinal drugs (Lok 2015) and regulating various body functions both in plant and animal kingdoms (Clemente *et al.* 2012; Oakley *et al.* 2008; Turnbaugh *et al.* 2009). Most of the microorganisms are very difficult to culture *in vitro*, but with the advent of Next Generation Sequencing (NGS) it has become easy to study the microbial world in their environments without culturing them *in vitro* (Scholz *et al.* 2012; Shokralla *et al.* 2012). NGS technology is a high throughput data sequencing technology which creates huge masses of data in less time (Metzker 2009). Different metagenomics tools and algorithms are available currently to process and analyse this massive amount of data. Next Generation Sequencing techniques have changed the perspective of viewing organisms, their relations and their phylogeny.

NGS has not only reduced the cost of sequencing but also the time because of its parallel nature. NGS technologies have also evolved over the time from Sanger sequencing followed by second generation systems like 454, Illumina, Ion Torrent and finally to third generation NGS technologies like Pacific Biosystems (Pac Bio) Single Molecule Real-Time (SMRT) sequencing and Oxford Nanopore Techniques (ONT) Nanopore Sequencing technologies (Allali *et al.* 2017; Bleidorn 2016). NGS, in general, and whole genome sequencing, in particular, are the major sources to the big data; a single NGS chip produces millions of reads in less time. This huge data has become a bottleneck not only for processing and storage but also for computational algorithms (Scholz, Lo and Chain 2012). This is one of the reasons for going to target sequencing rather than whole genome sequencing. Amplicon sequencing or target sequencing mainly uses a gene or portion of gene which has highly conservative and highly variable regions to differentiate between different levels of phylogeny. In metagenomics 16S rRNA gene is used for target sequencing, it is almost $\sim$1600

base pairs long and contains 9 hyper variable regions V1-V9, which are both variable as well as conserved (Janda and Abbott 2007).

Clustering or automatic classification has been an important tool in metagenomics used for taxonomic profiling. Clustering is binning of the data items based on the feature of the data; it has a lot of applications and usage in different fields (Prabhu and Duraiswamy 2013). Different algorithms have been published from time to time to explore the microbial environments (Bhat and Prabhu 2017). The short amplicon reads are clustered based on the similarity between each other in de-novo mode or with the reference database in closed-reference mode. Most algorithms use 97% similarity threshold for clustering to define the bacterial species (Schloss and Handelsman 2005; Schloss 2010). In metagenomics, clustering is done based on the similarity feature between the sequence reads with in the datasets or between the sequence reads of datasets and reference database. There are two different methods used to analyse the input NGS sequence data, clustering-first and assignment-first. (i) Clustering-first approach first clusters the sequence reads and then assigns them to the database as in Mothur (Schloss *et al.* 2009) and QIIME (Caporaso *et al.* 2010), (ii) while the latter uses reverse approach-first assignment and then clustering as in Kraken (Wood and Salzberg 2014) and CLARK (Ounit *et al.* 2015). The clustering-first approach can be closed-reference, open-reference or *de-novo*. The closed-reference method uses the database for clustering and clusters only the reads which hit the database. On the other hand, a *de novo* approach does not need a database but it matches the sequence reads within the given dataset itself, while the open-reference approach does the combination of both the closed-reference and *de-novo* approaches. The problem with assignment-first and closed approaches is that only those microbes which are annotated in the database can be found, while the novel and new species cannot be identified. Since only a small fraction of the microbial diversity has been explored, it is better to go for de-novo or open reference approaches in case of unknown environments (Westcott and Schloss 2015). In the present study all the algorithms use clustering first algorithms and *de-novo* approach for clustering the sequence reads into clusters.

The latter half of the paper is organised in the following way. Section 2 discusses background knowledge of the study. Section 3 discusses material, methods and validation metrics. Section 4 is about the results and discussion. Finally, section 5 concludes the paper.

2. Background

Metagenomics can use two categories of methods, whole genome shotgun sequencing (WGS) methods and target metagenomics methods, which are also called as amplicon sequencing. Shotgun sequencing takes the whole genome of the organism and provides rich information to explore the microbial community both functionally and taxonomically. But the problem with the shotgun sequencing is that it is very expensive as compared to the target metagenomics and is also computationally challenging and complex. On the other hand, target metagenomics uses only 16SrRNA gene, which makes it less expensive both computationally and cost-wise. Different OTU clustering algorithms have been published from time to time to analyse the metagenomics data, and a number of studies have been done to analyse these metagenomics algorithms (Chen *et al.* 2013; Quan *et al.* 2018). Various algorithms along with their qualitative parameters are tabulated in table 1. Most of these algorithms use two types of clustering approaches either greedy heuristics or hierarchical clustering approach, and a few use the model based clustering approaches.

Greedy based methods are partitioned clustering methods which select a sequence read as a seed and maps it either against other reads or against the database at a particular threshold value generally 97%, if the reads match the seed then they are grouped otherwise the read acts as a new seed. The main algorithms which use greedy heuristic approach are CDHIT (Li and Godzik 2006), USEARCH (Edgar 2010), UCLUST (Edgar 2010), VSEARCH (Rognes *et al.* 2016), SUMACLUST (Mercier *et al.* 2013), OTUCLUST (Albanese *et al.* 2015) GramCluster (Russell *et al.* 2010) and DNACLUST (Ghodsi, Liu and Pop 2011). CDHIT sorts all the sequence reads and keeps the longest read as its first seed and then bins the sequences which are similar to the seed at some threshold, if sequence is not similar then it acts as a new seed. UCLUST works in similar way as CDHIT but it does not use the longest read as a seed. VSEARCH takes advantage of Single Instruction Multiple Data (SIMD) parallelism and multiple threading to perform alignments at a high speed and uses an optimal global aligner. USEARCH uses a heuristic seed and extends aligner for the alignment search. SUMACLUST and OTUCLUST perform an exact sequence alignment and the clusters are constructed incrementally by comparing an abundance-ordered list of input sequences against the representative set of already-chosen sequences. GramCluster uses a grammar-based distance metric to cluster the sequence reads. DNACLUST uses a novel k-mer filtering algorithm instead of pairwise alignment method. Most of these greedy methods perform in $O(n)$ time complexity but cluster quality is reduced as compared to hierarchical methods.

Hierarchical methods use pairwise genetic distance matrix created by comparing all the reads with each other mostly in agglomerative way. Most of these algorithms have $O(n^2)$ time complexity, where n is number of sequence read, which is a bottleneck for processing metagenomic big data. The main algorithms in this class are Mothur's nearest neighbour, average neighbour and furthest neighbour, and ESPRIT (Sun *et al.* 2009). Mothur is modified Dotur. An improved version of average neighbour called SLP was also given which can reduce the sequencing noise and impact of abundant sequences to reduce the number of OTUs. ESPRIT uses the

Table 1. OTU clustering algorithms based on different qualitative parameters

Category	Algorithm	Reference category	Time complexity	Handling big data	Metagenomic pipeline	Open source	Implementation	Release year
Greedy Heuristic Clustering Algorithms	CDHIT (Li and Godzik 2006)	*de novo*	O (n)	Yes	QIIME-1.9, Mothur-1.39	Yes	C++	2006
	USEARCH (Edgar 2010)	closed, *de novo*, open	O (n)	Yes	QIIME-1.9	No	C++	2010
	UCLUST (Edgar 2010)	closed, *de novo*, open	O (n)	Yes	QIIME-1.9	No	C++	2010
	VSEARCH (Rognes *et al.* 2016)	*de novo*	O (n)	Yes	Mothur-1.39, QIIME-1.9	Yes	C++	2016
	SUMACLUST (Mercier *et al.* 2013)	*de novo*	O (n)	Yes	QIIME-1.9	Yes	C	2014
	OTUCLUST (Albanese *et al.* 2015)	*de novo*	O (n)	Yes	MICCA-1.6.1	Yes	Python, C	2015
Hierarchical Clustering Algorithms	ESPRIT (Sun *et al.* 2009)	*de novo*	O (n^2)	No	Code	Yes	Perl	2009
	Nearest, average, and furthest neighbour in Mothur (Schloss *et al.* 2009)	closed, *de novo*	O (n^2)	No	Mothur-1.39	Yes	C++	2009
Model Based Clustering Algorithms	Swarm (Mahé *et al.* 2014)	*de novo*	O (nl)	No	QIIME-1.9	Yes	C++	2014
	Crop (Hao *et al.* 2011)	*de novo*	O (n^2/k)	Yes	Code	Yes	C++	2011

pairwise global alignment whereas Mothur uses the multiple sequence alignment tool MUSCLE (Edgar 2004) to compute the pairwise distance matrix.

Model based clustering methods implement probability methods like Gaussian mixture model and machine learning techniques. Crop (Hao *et al.* 2011) implements unsupervised Bayesian clustering method to find the clusters without threshold. Swarm (Mahé *et al.* 2014) addresses the arbitrary global threshold by using a local threshold and clustering identical reads iteratively and then uses the abundance and internal structure of the cluster to optimize the results.

In the present study, five OTU clustering algorithms are being used for analysis; out of these Mothur's average neighbour (2009) is a hierarchical method, Swarm is a two phased agglomerative single-linkage-clustering algorithm while CDHIT (2006), SUMACLUST (2014) and UCLUST (2010) are greedy heuristic based methods. The greedy approaches create the clusters in the incrementing way based on the abundance but in case of SUMACLUST an exact sequence alignment is performed instead of heuristics as in CDHIT and UCLUST. Mothur is actually a pipeline which implements many algorithms like single linkage, average linkage, complete linkage, OptiClust and Vsearch. Swarm implements unsupervised single-linkage-clustering and builds clusters first by agglomerating similar sequence-reads and then uses the abundance for internal structure of OTU clusters. All the above mentioned algorithms perform *de novo* clustering, while UCLUST can also perform closed reference and open reference type of clustering also. The *de novo* approach performs better than closed (Westcott and Schloss 2015).

3. Materials and methods

3.1 *Datasets*

In this study, total number of datasets used is 14 (4 oral, 5 soil and 5 simulated) which are divided into three categories as shown in table 2.

3.1.1 *Soil microbiome (synthetic datasets):* Soil microbiome are Pyrosequencing datasets, generated by Roche 454sequencer, obtained from European Bioinformatics Institute (*http://www.ebi.ac.uk/ena/data/view/ERP001958*) under accession number ERP001958.The microbial dataset has

Table 2. Datasets used for the study

Datasets	Samples Ids	Seq-reads	Read length	Total reads
Oral Microbiome Illumina MiSeq Datasets	S1	25035	140–150	128545
	S2	39110		
	S3	41055		
	S4	23345		
Soil Microbiome Roche-454 Datasets	ERR193622	9302	200–500	44578
	ERR193623	6332		
	ERR193624	10619		
	ERR193625	9358		
	ERR193630	8967		
Simulated Microbiome (Grinder) Greengenes Datasets	gd_1000	1000	120–150	10000
	gd_1500	1500		
	gd_2000	2000		
	gd_2500	2500		
	gd_3000	3000		

been taken from a diesel-contaminated railway site to check the relationship between the microbial diversity, pollution level and soil physiochemical properties. The total number of reads is 445788, which are divided into 5 subsets. The datasets are pre-processed for quality check, filtering, and trimming by using the split-library command in QIIME. The read length is between 200bp and 500bp with an average GC content of 55% and Phred score >34.

3.1.2 *Oral microbiome:* In the presence of dentist concerned, oral swab samples are collected from caries individuals after getting informal consent. Samples are collected from caries affected patients who reported in K.S. Rangasamy Dental Science and Research Tiruchengode, Tamil Nadu. The swab samples are stored in XPBS. Metagenomics DNA has been extracted from the swab samples using Qiagen DNA microbiome kit protocol suit. Metagenome has been amplified using V3-V4 region of 16S rRNA gene using specific primers. The amplified metagenomics library has been subjected to Next Generation Sequencing using Illumina Nextera Platform. The datasets are under accession number SRP156445 in NCBI website. The datasets have been pre-processed for quality check, filtering, and trimming by using the split-library command in QIIME. The total number of reads after pre-processing is 128545, the range of read length is between 140bp and 150bp with average GC content of 53% and Phred score >32.

3.1.3 *Simulated microbiome:* Simulated dataset are obtained through Grinder 0.5.4 simulator by using Greengenes unaligned reference database (*ftp://greengenes.microbio.me/greengenes_release/gg_13_5/gg_13_8_otus.tar.gz*). Grinder is an open-source bioinformatics tool to simulate amplicon and shotgun genomics, metagenomics datasets from the reference sequences (Angly *et al.* 2012). As it is difficult to measure the performance of complex microbial species due to their diversity and abundance, simulated data sets are being used because of the known species abundance. To make the simulated data to look like natural, different complexities like read length and number of species are being considered for the datasets. Diversity is the number of full-length 16SrRNA reference sequences taken randomly to create a dataset. Diversity values ranging between 1000 and 3000 species are being used. Fold coverage is the number of base pairs in the output simulated data divided by the total number of base pairs in the input reference database. Coverage is controlled by using desired number of reads in this experiment. Errors can be mutations, homopolymers or chimeras in case of raw data sets. But in this case most of the data is cleaned with very few errors, hence it is assumed that 90% of sequencing and PCR errors are eliminated during the pre-processing stage. A total of 5 datasets are generated with different species number as gd_1000, gd_1500, gd_2000, gd_2500, gd_3000. All these datasets are created with different species number, and the overall total number of reads taken is 10000 and the read length is between 120 and 150bp with an average GC content of 55%.

In metagenomics microbiome datasets, the prior knowledge of distribution of species is difficult to know, as there is no golden reference database available for metagenomics data analysis. In order to get the ground truth, sequence reads are mapped against the annotated database (McDonald *et al.* 2012) using BLAST alignment tool (Altschul *et al.* 1990) at 97% identity to retain the ground truth. The procedure to generate the ground truth is depicted in figure 1. In this approach the taxonomic information headers are added to the sequences of Greengenes reference database by using TaxCollector tool (Giongo *et al.* 2010). The TaxCollector uses files from NCBI containing the taxonomy information, called names.dmp and nodes.dmp files. A new annotated database is created with each sequence having taxonomic information added to its header. The datasets are blasted against the annotated database. The output of the BLAST operation is filtered at 97% similarity to act as the ground truth (Woese 1987), (Sun *et al.* 2012) and (Cai and Sun 2011).

3.2 *Evaluation and assessment for clustering quality*

Comparing the results of a cluster analysis with the externally known results or externally given class labels and evaluating how well the results of a cluster analysis fit the data without reference to external information, various methods have been given. Some of the important methods mostly used in metagenomics OTU clustering methods are as under:

3.2.1 *Normalized mutual information:* Normalized mutual information (NMI) (Press et al. 2007) is measure of validation for clustering. It takes values between 0 and 1, where value 1 shows that the clusters created are same as that of golden reference while 0 show clusters are created randomly. Let there be N reads from m species (S_1, S_2, S_3, …, S_m) clustered into n clusters (C_1, C_2, C_3, …, C_n) at default similarity threshold of each algorithm. NMI can be defined as:

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

$$i = 1,2,\ldots m; \quad j = 1,2,\ldots,n$$

$$H(S) = -\sum_{i=1}^{m}\frac{S_i}{N}\log\frac{S_i}{N}$$

$$H(C) = -\sum_{j=1}^{n}\frac{C_j}{N}\log\frac{C_j}{N}$$

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

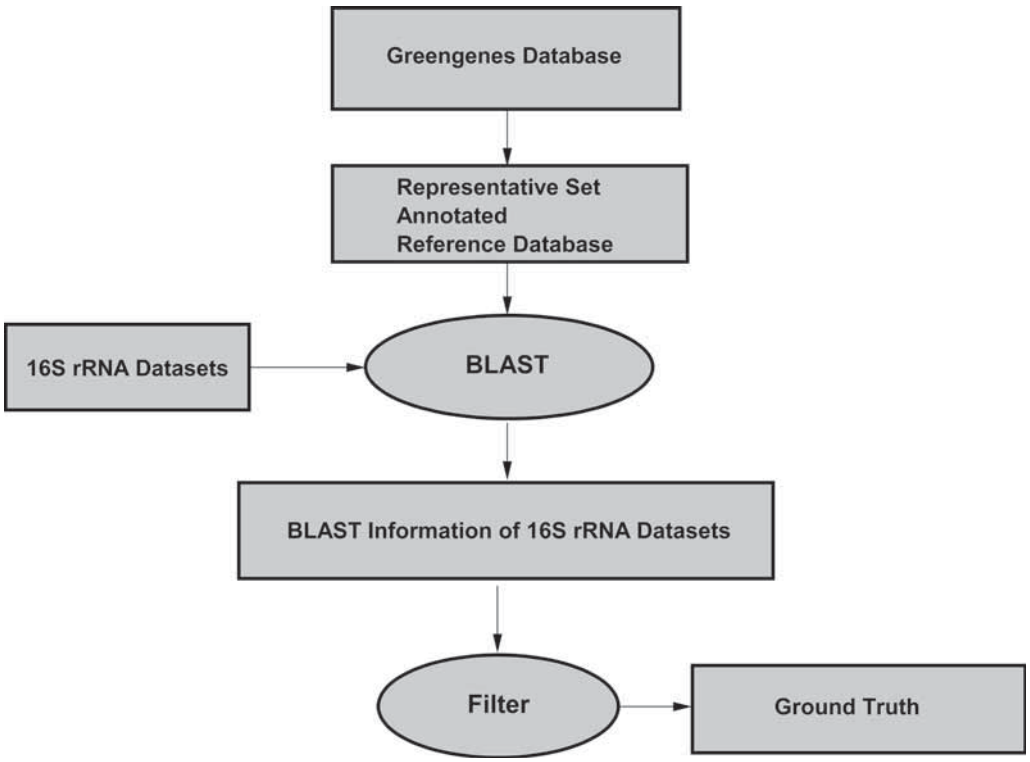

Figure 1. Procedure to generate ground truth for 16S rRNA datasets.

Where I is the mutual information between m species (S) and n clusters and H is entropy.

3.2.2 *Normalized information difference:* Normalized information difference (NID) (C. J. Van Rijsbergen 1979) is also a clustering validation technique. It takes values between 0 and 1, smaller NID values imply better clustering results. NID shows tighter bound when compared to NMI and it is a useful cluster validation method. NID is defined as:

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

3.2.3 *Precision and recall:* Precision p_{ij} is the number of sequence reads present in both class i and cluster j, divided by the read number in cluster j, so it defines the homogeneity of cluster j. Recall r_{ij} is the proportion of sequence reads from class i present in cluster j, so it defines the completeness. Let $|S_i|$ is the true read number from species i, $|C_j|$ is the read number from cluster j, and a_{ij} is read number from i species and binned into j cluster. Precision and Recall is defined as:

$$p_{ij} = \frac{a_{ij}}{|C_j|}$$

$$r_{ij} = \frac{a_{ij}}{|S_i|}$$

All the experiments have been done on Linux Mint 18.3 Cinnamon 64-bit machine with Intel core i3-4130 processor and 8 GB RAM and all the OTU clustering algorithms are executed on default identity threshold values.

4. Results and discussion

4.1 *Inferred number of OTUs*

In this study five OTU clustering algorithms are being examined as shown in the table 3. In oral microbiome Mothur's average method has yielded the highest number of OTUs followed by Swarm, CDHIT, and UCLUST; SUMACLUST has generated the lowest number of OTUs. In soil microbiome Mothur's average neighbour created the highest number of OTUs followed by Swarm, UCLUST, and CDHIT. Again SUMACLUST has obtained the lowest number of OTUs but in soil microbiome UCLUST got more OTUs than CDHIT as unlike in Oral microbiome. In case of simulated datasets again Mothur created the highest number of OTUs followed by Swarm; whereas CDHIT generated more number of OTUs than the UCLUST; SUMACLUST got the lowest of all. Most of the clustering methods apply post-processing procedure to eliminate singletons or doubletons, but to evaluate the performance metrics such as NMI, NID and Precision-Recall, singletons and doubletons should be retained (Park *et al.* 2018). This is the main reason that all the methods produce greater number of OTUs. Most of these overestimated OTUs are singletons only. Except

Table 3. Number of OTU clusters created by each algorithm

Datasets	Sample Ids	Seq reads	Algorithms				
			Mothur (Avg.)	Swarm	UCLUST	SUMACLUST	CDHIT
Oral Microbiome Illumina MiSeq Datasets	S1	25035	20710	18517	11702	10869	12428
	S2	39110	31263	26317	15903	14149	16771
	S2	41055	33482	30761	20084	18626	21236
	S4	23345	19248	17013	10893	10106	11663
Soil Microbiome Roche-454 Datasets	ERR193622	9302	8744	7494	1745	1151	1744
	ERR193623	6332	6136	5779	2096	1367	2014
	ERR193624	10619	8637	10497	5451	3796	5353
	ERR193625	9358	9162	8651	2892	1827	2669
	ERR193630	8967	8914	8160	2828	1776	2790
Simulated Microbiome (Grinder) Greengenes Datasets	gd_1000	1000	999	999	999	983	999
	gd_1500	1500	1499	1499	1498	1481	1497
	gd_2000	2000	1996	1995	1992	1962	1993
	gd_2500	2500	2496	2493	2487	2456	2487
	gd_3000	3000	2987	2985	2979	2937	2981

Swarm, all other algorithms are executed on 97% similarity threshold for the cluster formation (Schloss and Handelsman 2005; Schloss 2010). In all the three data categories of oral, soil and simulated data, the Mothur's average neighbour algorithm and Swarm created more number of clusters than greedy clustering algorithms.

4.2 *Accuracy comparison*

For validating and comparing the performance of OTU clustering algorithms besides calculating the number of inferred OTUs, NMI and NID have been calculated, as shown in figure 2. After obtaining the values of NMI and NID for fourteen datasets from three environmental categories i.e. oral microbiome, soil microbiome and simulated data for each algorithm; mean and standard deviation have been calculated. Higher NMI values are obtained from Mothur's Average-Neighbour (0.6602), followed by Swarm (0.6597). In greedy algorithms CDHIT (0.4513) showed better results than UCLUST (0.4492) and SUMACLUST (0.4108).

Both the Mothur's hierarchical algorithm and Swarm showed better performance according to NMI values than their greedy heuristic sister algorithms. In terms of NID again Mothur's Average-Neighbour (0.3452) got the lowest value followed by Swarm (0.3490). In greedy heuristic methods CDHIT (0.5735) showed the lowest values followed by UCLUST (0.5747) and SUMACLUST (0.6099). From the metrics of NMI and NID values it can be observed that Mothur's Average method and Swarm performed better than greedy heuristic clustering algorithms. In addition to NMI and NID validations to assess properly the relation of OTU clusters to the community structure, Precision and Recall have also been calculated to get the more information on how the sequence reads from similar species cluster

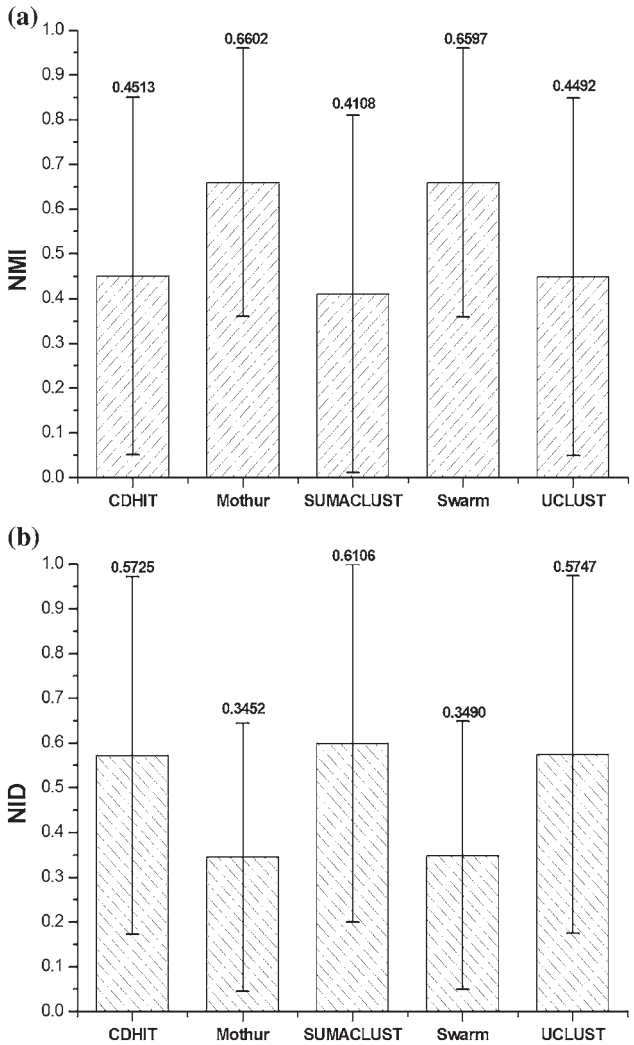

Figure 2. (**a**) Normalized mutual information (NMI) and (**b**) normalized information difference of five OTU Clustering Methods.Average mean scores of the datasets are reported together with theerror bars.

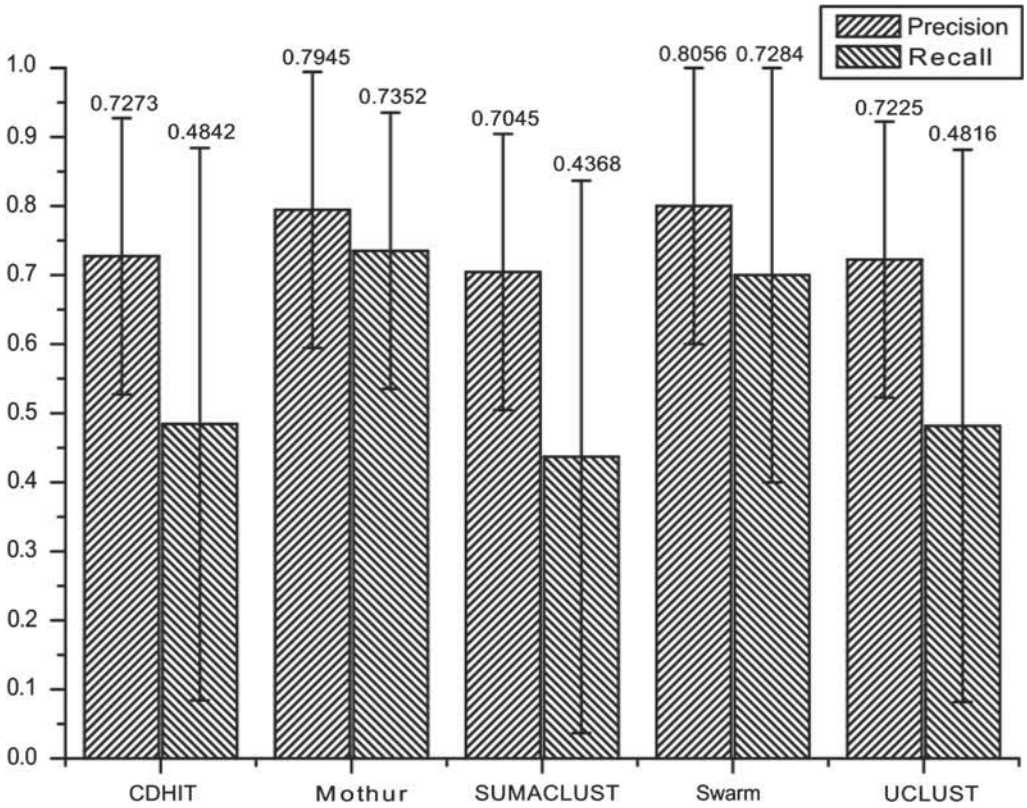

Figure 3. Precision and Recall of five OTU Clustering Methods. Average mean scores of the datasets are reported together with the error bars.

together and how the sequence reads are distributed among the OTU clusters by different OTU clustering algorithms as shown in figure 3.

Both precision and recall values are average mean of oral microbiome, soil microbiome and simulated datasets, and the error bars represent the deviation. Swarm showed the highest precision score (0.8056) followed by Mothur's Average-Neighbour (0.7945) while as SUMACLUST showed the lowest precision score (0.7045). CDHIT (0.7273) and UCLUST (0.7225) have shown almost similar results for Precision Score. In case of recall, Mothur's Average-Neighbour (0.7352) has obtained the highest value followed by Swarm (0.7284), CDHIT (0.4842) and UCLUST (0.4816), While SUMACLUST (0.4368) has obtained the lowest recall value.

4.3 *Time demand of the algorithms*

Time complexity is one of the important factors in analysis, the average computational time taken by the algorithms is depicted in table 4. UCLUST (0.631×10^1s) and CDHIT (0.237×10^2s) have the fastest running times followed by Swarm (0.284×10^2s) and SUMACLUST (0.925×10^3s), While Mothur (0.228×10^5s) has the slowest running time. Out of non-greedy algorithms, Swarm performed better than that of Mothur's Average-Neighbour and in greedy clustering algorithms UCLUST performed better than others. In general, the results show that in terms of cluster qualities from NMI, NID, Precision and Recall values, the Swarm and hierarchical OTU clustering algorithm performed well, whereas in terms of time, greedy approaches particularly UCLUST and CDHIT performed well. So it is necessary to take into account all of these things while performing the analysis of huge amount of data.

Table 4. Average computational time of all the five clustering algorithms

S/No.	Algorithms	Average real time (s)
01	CDHIT	0.237×10^2
02	Mothur (Avg. Neighbour)	0.228×10^5
03	SUMACLUST	0.925×10^2
04	Swarm	0.284×10^2
05	UCLUST	0.631×10^1

5. Conclusion

Next generation sequencing methods have created huge quantity of data from target metagenomics and shotgun sequencing. This has necessitated the creation of parallel tools that could handle and analyse this ocean of big data

efficiently and smoothly. In the past it had been a trend to use reference dependent algorithms to analyse the data. But the problem with these methods is that only those microbes whose sequence reads get hit from the database can be analysed while most of the microbes remain unknown, and there are very less chances of getting novel species. Taxonomy independent methods have filled that gap. But in this case the OTU clusters created have an important impact on downstream processing. It is difficult to arrive at the best method when the available information is less. In the present study data samples from oral microbiome Illumina datasets, soil microbiome

Pyrosequencing 454 datasets and simulated datasets have been taken for a critical analysis and comparison of the performance of the current five state-of-the-art OTU clustering algorithms. From the results we got Swarm and Mothur's Average Neighbour method showed better quality clusters in all the three categories of datasets than greedy based approaches but in terms of time complexity, greedy heuristic based methods showed better performance. In the field of Bioinformatics, there are numerous methods and software packages available to perform the task of clustering. Even for an experienced researcher it becomes difficult to choose the correct method for their study. The present study provides a view of comparison between the most popularly used algorithms in Metagenomics for performing clustering of16SrRNAshort sequence read data. The study may be used as a resource for selecting a specific method for clustering and analysing long sequence reads as well as short sequencing reads. So finally, based on the amount of data to be analysed, read length of datasets and experimental setup to be used one should consider all of these things before analysing the data.

Acknowledgements

The authors are grateful to the Management and Principal of K.S.Rangasamy College of Technology, Tiruchengode, Namakkal, Tamil Nadu, India, and DST FIST (Fund for Infrastructure for Science and Technology) FIST No: 368 (SR/FST/College – 235/2014)] and DBT-STAR College Scheme (BT/HRD/11/09/2018) for proving laboratory support in the Institution. Also, we would like to thank Prof. M. Umaiarasi for proof-reading of the manuscript.

References

Albanese D, Fontana P, De Filippo C, Cavalieri D and Donati C 2015 MICCA: A complete and accurate software for taxonomic profiling of metagenomic data. *Sci. Rep*. **5** 1–7

Allali I, Arnold JW, Roach J, *et al.* 2017 A comparison of sequencing platforms and bioinformatics pipelines for compositional analysis of the gut microbiome. *BMC Microbiol*. **17** 194

Altschul SF, Gish W, Miller W, Myers EW and Lipman DJ 1990 Basic local alignment search tool. *J. Mol. Biol.* **215** 403–410

Angly FE, Willner D, Rohwer F, Hugenholtz P and Tyson GW 2012 Grinder: A versatile amplicon and shotgun sequence simulator. *Nucleic Acids Res*. **40** e94

Bernard G, Pathmanathan JS, Lannes R, Lopez P and Bapteste E 2018 Microbial dark matter investigations: how microbial studies transform biological knowledge and empirically sketch a logic of scientific discovery. *Genome Biol. Evol.* **10** 707–715

Bhat AH and Prabhu P 2017 OTU clustering: A window to analyse uncultured microbial world. *Int. J. Sci. Res. Comput. Sci. Eng.* **5** 62–68

Bleidorn C 2016 Third generation sequencing: Technology and its potential impact on evolutionary biodiversity research. *Syst. Biodiver*s **14** 1–8

Cai Y and Sun Y 2011 ESPRIT-Tree: Hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time. *Nucleic Acids Res.* **39** 1–10

Caporaso JG, Kuczynski J, Stombaugh J, *et al.* 2010 QIIME allows analysis of high-throughput community sequencing data. *Nat. Methods* **7** 335–336

Chen W, Zhang CK, Cheng Y, Zhang S and Zhao H 2013 A Comparison of methods for clustering 16S rRNA sequences into OTUs. *PLoS One* **8** e70837

Clemente JC, Ursell LK, Parfrey LW and Knight R 2012 The impact of the gut microbiota on human health: An Integrative view. *Cell* **148** 1258–1270

Edgar RC 2004 MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res*. **32** 1792–1797

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

Ghodsi M, Liu B and Pop M 2011 DNACLUST: Accurate and efficient clustering of phylogenetic marker genes. *BMC Bioinformatics* **12** 271

Giongo A, Davis-richardson AG, Crabb DB and Triplett EW 2010 TaxCollector: Modifying current 16S rRNA databases for the rapid classification at six taxonomic levels. *Diversity* **2** 1015–1025

Hao X, Jiang R and Chen T 2011 Clustering 16S rRNA for OTU prediction: A method of unsupervised Bayesian clustering. *Bioinformatics* **27** 611–618

Janda JM and Abbott SL 2007 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: Pluses, perils, and pitfalls. *J. Clin. Microbiol*. **45** 2761–2764

Kellenberger E 2001 *Exploring the unknown. EMBO Rep* (Vol. 2) (Oxford, UK: Wiley Online Library)

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

Lok C 2015 Mining the microbial dark matter. *Nature* **522** 270–273

Mahé F, Rognes T, Quince C, de Vargas C and Dunthorn M 2014 Swarm: robust and fast clustering method for amplicon-based studies. *PeerJ* **2** e593

McDonald D, Price MN, Goodrich J, *et al.* 2012 An improved Greengenes taxonomy with explicit ranks for ecological and evolutionary analyses of bacteria and archaea. *ISME J.* **6** 610–618

Mercier C, Boyer F, Bonin A and Coissac E 2013 SUMATRA and SUMACLUST: fast and exact comparison and clustering of sequences. Abstr SeqBio 25-26th Nov 2013 27

Metzker ML 2009 Sequencing technologies – the next generation. *Nat. Rev. Genet*. **11** 31

Oakley BB, Fiedler TL, Marrazzo JM and Fredricks DN 2008 Diversity of human vaginal bacterial communities and associations with clinically defined bacterial vaginosis. *Appl. Environ. Microbiol.* **74** 4898–4909

Ounit R, Wanamaker S, Close TJ and Lonardi S 2015 CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. *BMC Genomics* **16** 1–13

Park S, Choi H, Lee B, Chun J, Won J and Yoon S 2018 hc-OTU: A fast and accurate method for clustering operational taxonomic units based on homopolymer compaction. *IEEE/ACM Trans. Comput. Biol. Bioinforma* **15** 441–451

Prabhu P and Duraiswamy K 2013 An efficient visual analysis method for cluster tendency evaluation, data partitioning and internal cluster validation. *Comput. Informatics* **32** 1013–1037

Rognes T, Flouri T, Nichols B, Quince C and Mahé F 2016 VSEARCH: a versatile open source tool for metagenomics. *PeerJ* **4** e2584

Russell DJ, Way SF, Benson AK and Sayood K 2010 A grammar-based distance metric enables fast and accurate clustering of large sets of 16S sequences. *BMC Bioinformatics* **11** 601

Schloss PD 2010 The effects of alignment quality, distance calculation method, sequence filtering, and region on the analysis of 16S rRNA gene-based studies. *PLoS Comput. Biol.* **6** 19

Schloss PD and Handelsman J 2005 Introducing DOTUR, a computer program for defining opera: onal taxonomic units and Es:ma:ng species richness. *Appl. Environ. Microbiol.* **71** 1501–1506

Schloss PD, Westcott SL, Ryabin T, *et al.* 2009 Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl. Environ. Microbiol.* **75** 7537–7541

Scholz MB, Lo CC and Chain PSG 2012 Next generation sequencing and bioinformatic bottlenecks: The current state of metagenomic data analysis. *Curr. Opin. Biotechnol*. **23** 9–15

Shokralla S, Spall JL, Gibson JF and Hajibabaei M 2012 Next-generation sequencing technologies for environmental DNA research. *Mol. Ecol.* **21** 1794–1805

Sun Y, Cai Y, Huse SM, Knight R, Farmerie WG, Wang X and Mai V 2012 A large-scale benchmark study of existing algorithms for taxonomy-independent microbial community analysis. *Brief Bioinform*. **13** 107–121

Sun Y, Cai Y, Liu L, Yu F, Farrell ML, McKendree W and Farmerie W 2009 ESPRIT: estimating species richness using large collections of 16S rRNA shotgun sequences (supplementary data). *Nucleic Acids Res*. **39** 1–18

Turnbaugh PJ, Hamady M, Yatsunenko T, *et al.* 2009 A core gut microbiom in obese and lean twins. *Nature* **457** 480–484

Van Rijsbergen CJ 1979 *Information retrieval* (2nd ed.) (Butterworth-Heinemann Newton, MA, USA)

Westcott SL and Schloss PD 2015 De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. *PeerJ* **3** e1487

William H.Press, Saul A. Teukoisky, William T. Vetterling BPF 2007 *NUMERICAL RECIPES The Art of Scientific Computing* (Cambridge University Press)

Woese CR 1987 Bacterial evolution. *Microbiol. Rev.* **51** 221–71

Wood DE and Salzberg SL 2014 Kraken: Ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.* **15** R46

Zou Q, Lin G, Jiang X, Liu X and Zeng X 2018 Sequence clustering in bioinformatics: an empirical study. *Brief. Bioinformatics* bby090, *https://doi.org/10.1093/bib/bby090*

Corresponding editor: BJ Rao