**A Comprehensive Review on Metagenomic OTU Clustering Algorithms**

**\*1Ashaq Hussain Bhat**

[**\*1ashaq11bhat@gmail.com**](mailto:*1ashaq11bhat@gmail.com)

**2Hashmat Fida**

[**Hasmatfida@gmail.com**](mailto:Hasmatfida@gmail.com)

**3Danish Rashid Pala**

[**danishpala.cs@gmail.com**](mailto:danishpala.cs@gmail.com)

**4Maajid Bashir**

[**maajid.net@gmail.com**](mailto:maajid.net@gmail.com)

**\*1,3,4 Institute of Computing, Chandigarh University, Punjab, India-140413**

**2CSE Department Chandigarh University, Punjab, India-140413**

**Corresponding Auther:**[\*1ashaq11bhat@gmail.com](mailto:*1ashaq11bhat@gmail.com),

**Abstract—** The massive amount of data spawned from parallel High Throughput Metagenomics Next Generation Sequencing (mNGS) is a dare task to current technology for processing and analysing due to its enormous scale and computational complexity. Clustering is a crucial step in analysing large Next Generation Sequencing (NGS) datasets to determine microbial community compositions, taken from different environmental sites. Clustering the data before analysis and assembling offers a unique opportunity to analyse the massive data in depth. One of the significant yields of metagenomic data analysis is taxonomic profiling. It is done by exploiting the presence of hyper-variable regions (1-9) in the 16S rRNA gene. The clustering tools in metagenomics use the similarity feature of gene sequences for grouping, such groups are called as Operational Taxonomic Units (OTUs).Each clustering tool has different way of grouping the sequence reads. The OTU clustering algorithms either produce good quality clusters at the cost of time or decrease time complexity at the cost of cluster quality. In this paper we have given a comprehensive review on various metagenomic OTU clustering algorithms along with their working and technical details.

**Keywords: Metagenomic Clustering, NGS, taxonomic profiling, OTUs, microbial communities**

1. **Background**

The field of Metagenomics deals with the process of metagenome generation, which encompasses various stages like genome extraction, sequencing followed by grouping and mapping of sequences to a database reference and finally genome annotation. It is a molecular biological tool used to examine DNA recovered directly from the various environmental sites in order to research microbial communities existing within the samples, without necessity of acquiring unmixed cultures. The in-depth genomic analysis of uncultured microbes and their interactions with surrounding environmental conditions related to some specific functions is done using functional metagenomics(Marić et al., 2024). The mechanism of metagenomics is becoming familiar in-extensive genomic applications as a way to research the taxonomic and functional composition of microbial sections from analysing environmental, agricultural and clinical studies(Kalpana et al., 2020) (Guinane & Cotter, 2013)(Hattori & Taylor, 2009). Handelsmann et al, first introducer of Metagenomics in 1988, introduced the idea of analysing soil microbiome without using *in vitro* culturing by examining microbial genome directly by using *in silico* methods (Handelsmanl et al., 1998)(Gilbert & Dupont, 2011). As the sample data are excerpted directly from spots under survey without reliance on microbial culture, it inspires the studies of complex cultural microbes inside the laboratories by saving time and cost for the collecting fragments. Reporting survey, 99% of microorganisms are not culturable while as only 20% of human Intestinal microbiota are culturable according to another survey(Eckburg et al., 2005)(Nelson, 2011). Microbial ecology mostly focuses on microbial diversity, abundancies and microbial activity. Under diversity falls different sub-diversities like phylogenetic, species, genotype, gene, evolutional, metabolic and functional diversities (XU, 2006). The *in silico* analysis of metagenomic data has revealed tremendous amount of potential applications for human life, medicine, environment and other living beings. These features have driven more in-depth research in microbial diversity using *in silico* analysis and such analysis are impossible using *in vitro* methods. There are few of the metagenomic projects that became big source of contribution to the research and are globally accessible and freely available to everyone, the important of them are Earth Microbiome Project (Thompson et al., 2017) and (Human Microbiome Jumpstart Reference Strains Consortium et al., 2010)(Roy et al., 2024).

The expeditious growth from the past decade in metagenomics is closely related with huge progress in sequencing technologies. The just data collection to directly processing massive volumes of sequencing data without going to *in vitro* laboratory work, is the huge evolution of Next Generation sequencing (NGS) in metagenomic research. Furthermore, the boost and bloom of research experiments in metagenomic projects allowed ease of sequencing, collection, binning and overall data depiction with the help and advent of Third Generation Sequencing TGS). (Metzker, 2009) (Pareek et al., 2011). Annually, thousands of metagenomic studies are being conducted however in 2006 alone only two metagenomic studies had been conducted. In addition day by day new data is added to references databases and they are becoming larger and larger each day. For example, GenBank database of National Centre for Biotechnology Information (NCBI) contains more than 100,000 prokaryotic genomes at present which had covered only up to 300 prokaryotic genomes in the year 2006, (Levy & Myers, 2016)(Bansal et al., 2018).

Clustering Operational Taxonomic Units (OTUs) is a crucial activity in metagenomic research because it involves grouping related sequences and helps the researcher gather an insight into the diversity of microbial communities involved. The precision and efficiency of OTU clustering methods directly affect the downstream analyses and consequent conclusions drawn regarding microbial ecology. However, there currently exist OTU clustering methods that represent a trade-off between clustering quality and computational efficiency. Higher quality methods are often too slow to be practical for large datasets, while faster methods sacrifice clustering accuracy and may therefore lead to incorrect biological interpretations. It is therefore necessary to evaluate and analyze the various OTU clustering methodologies so that one may advise researchers thus selecting an algorithm that suits their aims-their accuracy versus efficiency-for their sequencing application. This study heavily emphasizes the need for discerning the technical features, advantages, and disadvantages of the different OTU clustering techniques used in metagenomic research. This paper reviews various OTU clustering algorithms: their working principles, technical details, and performance characteristics. It discusses the time complexity and the clustering quality trade-off and elaborates upon their impact in various scenarios. The paper provides a clear classification syntax based on clustering approaches that a researcher might find useful. A handy guide is prepared for researchers in selecting the most appropriate algorithm according to their specific metagenomic datasets and research goals.

1. **CLUSTERING**

The process of defining groups from the unlabelled data without the external knowledge of group labels is called unsupervised learning or clustering. It categorizes the data instance into subgroups in such a way that the similar instances are assembled together in one subgroup and the dissimilar instances are put in different subgroups (Amelio & Tagarelli, 2019). The purpose of clustering is to find the natural arrangements within the data which are alike together but unalike from other clusters. The structure of clustering is represented as a set of subsets, Cl=cl1, cl2… clk of dataset DS, such that: DS=Cli and Cli∩Clj=Ø for i≠j, afterwards any instance in DS belongs to exactly one and only one subset. In the perspective of data-mining field that deals with massive magnitude of raw data, the process of clustering is repeatedly used for various applications so that the minor quantity of date could be used for performing a more expensive analysis later(Marić et al., 2024; Roy et al., 2024). In the context of metagenomics, clustering approach is used to assign DNA sequence into clusters whom we call as Operational Taxonomy Units (OTUs) or phylotypes for estimating the diversity of microbiome community(Janda & Abbott, 2007). Algorithm, the step by step procedure that normally takes some input after accessing gives some output. It can be presumed as the instructions of a calculating appliance, little complicated and convoluted. The steps can be sequential, parallel or both. Clustering algorithms returning the output in the form of clusters by taking the data as input, such clusters can be self-reliant partitions or a hierarchy representing some sort of connections among the data, from which a data division can be obtained by hierarchy cutting at a significant degree (Xu & Tian, 2015). Clustering can be generally hard or soft. The clustering which allows every data point to belong to only group is known as hard clustering while the clustering which allows the data points to have some relation or probability to more than one cluster is Soft Clustering (Aggarwal & Reddy, 2014). Traditional clustering methods can be splitted in two clustering i.e. partitioned and hierarchical clustering, but there has been different taxonomies based on objective functions or aiming at definite format desired for the derived clusters(Xu & Tian, 2015)(Olfa & N’Cir, n.d.)(Ullman, J.D. & N.d., n.d.)(MacCuish et al., 2010). The figure 1.1 shows various types of clustering methods along with examples.

1. **OTU Clustering**

The volume and complexity of microbial world is very high, because of its complex nature most of it is still undiscoverable and many of their applications are unknown, for this reason it is sometimes referred as the dark matter of biological world (Bernard et al., 2018; Lok, 2015). Generally the metabarcoding or metagenomics analysis is divided into three steps as: pre-processing of data, OTU clustering and downstream processing. The pre-processing step comprises demultiplexing of sequences from barcodes, filtering the noise, and error elimination tasks, while the downstream processing involves statistical analysis of data and visualizing the analyzed data. The most significant phase is the clustering phase, which has received lot of attention from last few years and is still a potentially active area of research(Hussain Bhat & Prabhu, 2017). The purpose of OTU clustering is to find the natural arrangement or clusters within the metagenomic sequence data, which are similar together but different from other clusters (Bhat & Prabhu, 2017; Fahad et al., 2014; Rodriguez et al., 2019). Clustering is implemented for taxonomic presentation of microbial communities by binning the 16S rRNA read sequences into the groups or Texas called as Operational Taxonomic Units (OTUs). In case of taxonomic profiling of microorganisms, 16S rRNA gene sequences and 9 hyper-variable regions of the gene are used, more specifically in the kingdom of Bacteria. The 16S rRNA gene is almost composed of ~1600 base pairs, it has 9 (V1-V9) regions which are hyper variable in nature (Janda & Abbott, 2007). A developing quantity of binning tools has been spread from last decade to cluster the amplicon reads into cluster of groups(Bhat et al., 2019). Generally these clustering algorithms can be distributed into 3 categories: database dependent approaches or closed reference approaches, database independent or *de novo* approaches and hybrid of former two called open referencing approaches (Navas-Molina et al., 2013). The database dependent algorithms map the amplicon sequence reads against the reference database and cluster the amplicon reads which are identical to the sequence existing within the database generally above some threshold. But the obstacle with the instant approaches is that only those amplicon reads are clustered that hit the database and those that do not are discarded, and there are ample chances in getting the novel microbial species. Self-Reliant Database *de novo* procedures do not require any database to search; instead these methods map the amplicon reads within the data itself. These methods have advantages over closed reference methods, like there are more chances of getting novel species and no sequence is discarded. Third category is hybrid which first uses database for searching then for those amplicon reads which do not hit database are handled in *de novo* way(Westcott & Schloss, 2015).

1. **OTU Clustering Approaches**

The OTU clustering depends on a number of parameters; among them clustering algorithm design and the sequence similarity degree are foremost. Clustering of massive amount of sequence reads contained within the metagenomic datasets is impossible without a proficient and effective algorithm. Using the brute force method is incredible and beyond the belief. Specialized algorithms which are quicker than prominent searching methods like BLAST(Altschul et al., 1990) had been proposed to metagenomic applications from the last few decades to handle NGS data. Mostly these algorithms use three approaches as (i) greedy heuristics approaches, (ii) hierarchical clustering approaches and (iii) model based approaches. Figure 1 depicts overview of various metagenomic OTU clustering types along with examples. The various amplicon sequencing OTU clustering algorithms like complexity, handling big data, release year are based on various qualitative variables which are thoroughly defined in Table 1.

C:\Users\Administrator\Desktop\reviewPaper-oct-19\otuClustering.tif

**Figure 1 Metagenomic OTU Clustering Algorithm Types and Examples**

**Table 1: Details of OTU Clustering Algorithm Types**

| **Algorithm** | **Category** | **Searching Operation** | **Source** | **Tool/Software package using** | **Time Complexity** | **Free Source** | **Implementation Language** | **Handling**  **Big Data** | **Year of Release** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| CDHIT | Greedy Heuristic Clustering Algorithms | *de novo* | (Li & Godzik, 2006) | QIIME-1.9, Mothur-1.39 | Linear  *O(n)* | Open | C++ | Yes | 2006 |
| VSEARCH | *de novo* | (Rognes et al., 2016) | Mothur-1.39, QIIME-1.9 | Linear  *O(n)* | Open | C++ | Yes | 2016 |
| UCLUST | open,  *de novo*, closed | (Edgar, 2010) | QIIME-1.9 | Linear  *O(n)* | Proprietary | C++ | Yes | 2010 |
| USEARCH | open,  *de novo*, closed | (Edgar, 2010) | QIIME-1.9 | Linear  *O(n)* | Proprietary | C++ | Yes | 2010 |
| OTUCLUST | *de novo* | (Albanese et al., 2015) | MICCA-1.6.1 | Linear  *O(n)* | Open | C, Python | Yes | 2015 |
| SUMACLUST | *de novo* | (Mercier et al., 2013) | QIIME-1.9 | Linear  *O(n)* | Open | C | Yes | 2013 |
| DNACLUST | *de novo* | (Ghodsi et al., 2011) | Code | Linear  *O(n)* | Open | C++ | Yes | 2011 |
| OptiClust | *de novo* | (Westcott & Schloss, 2017) | Mothur-1.39 | Quadratic  *O(n2)* | Open | R | Yes | 2017 |
| GramCluster | *de novo* | (Russell et al., 2010) | Code | Linear  *O(n)* | Open | C | Yes | 2010 |
| SortMeRNA | *closed* | (Kopylova et al., 2012) | QIIME-1.9 | Linear  *O(n)* | Open | C++ | Yes | 2012 |
| DMSC | *de novo* | (Z.-G. Wei & Zhang, 2019) | Code | *—* | Open | C++ | Yes | 2019 |
| HBEA | *de novo* | (Bhat et al., 2021) | Code | Quadratic  *O(n2)* | Open | Python | No | 2021 |
| ESPRIT | Hierarchical Clustering Algorithms | *de novo* | (Sun et al., 2009) | Code | Quadratic  *O(n2)* | Open | Perl | No | 2009 |
| ESPRIT-tree | *de novo* | (Cai & Sun, 2011) | Code | *O(n1.3)* | Open | Perl | No | 2011 |
| Nearest neighbor, average neighbor, & furthest neighbor in Mothur | *de novo,* closed | (Schloss et al., 2009) | Mothur-1.39 | Quadratic  *O(n2)* | Open | C++ | No | 2009 |
| Swarm v1 | Model Based Clustering Algorithms | *de novo* | (Mahé et al., 2014) | QIIME-1.9 | *O(nl)* | Open | C++ | No | 2014 |
| Swarm v2 | *de novo* | (Mahé et al., 2015) | QIIME-1.9 | *O(nl)* | Open | C++ | Yes | 2015 |
| Crop | *de novo* | (Hao et al., 2011) | Code | *O(n2/k)* | Open | C++ | Yes | 2011 |
| Sig Clust | *de novo* | (Chappell et al., 2018) | Code | Linear  *O(n)* | Open | C++ | Yes | 2018 |
| DMclust | *de novo* | (Z.-G. Wei et al., 2017) | Code | *O(n+m2+l\*m2+k\*m)* | — | — | Yes | 2017 |
| IFCM | *de novo* | (Liu et al., 2017) | Code | *—* | Open | C++ | Yes | 2017 |
| GeFast | *de novo* | (Müller & Nebel, 2018) | Code | *O(nl)* | Open | C++ | Yes | 2018 |

* 1. **Greedy heuristic approaches**

Greedy heuristic approach is an [algorithmic paradigm](https://en.wikipedia.org/wiki/Algorithmic_paradigm) that selects local optimal choice always at every level of stage with the purpose of finding the global optimum. Usually greedy strategies don’t produce an optimal solution despite a greedy heuristic yield optimal solution that approximates everywhere the optimal solution in appropriate time. Traditionally greedy heuristic clustering methods for biological sequence data generally require the guide tree construction which is consequently used for Multiple Sequence Alignment tool or (MSA) to each sequence (Ghodsi et al., 2011). But mostly such type of tools choose a seed i.e. sequence read and map it against other reads within the data or in a database at a specific value which we call threshold database value which is generally 97% as usual, a cluster is formed if the reads matches at or above this threshold otherwise the read is treated as the new seed. (Chen et al., 2013) as illustrated in Figure 2. Most of the greedy heuristic OTU clustering algorithms employ similar type of strategies with minimal differences in case of finding similarities among sequence reads, sorting of sequences, procedure for the steps of algorithm weather serial or parallel. Most of the tools can be customized like analyzing at different threshold options; in addition of having default options. Some tools give more importance to time and use k-mer filtering, while others emphasis on quality of clustering and use customized versions of pair wise global alignment. Input sequence reads are sorted by CDHIT and longest read is put as its initial seed or centroid for the cluster, then aligns each sequence with the seed and binds the sequence, which is identical to the seed cluster at a given threshold value, and if not alike, then it works like afresh cluster seed (Li & Godzik, 2006). CDHIT tool is using counting of short words and indexing table to ignore the alignments which are not useful. OpenMP is used to implement multithreading parallelized version of CDHIT, it exercise multithreading in such a way that one thread is taking care of read/write to and makes a indexing global table whilst others perform the process of clustering to sequences(Fu et al., 2012). USEARCH implements a method i.e. locally sensitive hash function and matches bi-sequences locally by individual words, called as k-mers instead of going for entire sequence (Edgar, 2010). The sequences having most of the k-mers common are selected for actual alignment. Centroid/seed is compared by USEARCH with target sequence in declining manner by using common rare points amidst the centroid sequence and the targeted sequence, and the list which is sorted is considered as "U vector". The process of searching concludes when the given number of accepts by default is 1 and refuses after the given number by default is 8, as overall same and unique words coordinate well with similarity. So there are chances of finding the cluster members in few searches that share common unique words, and this could increase the search speed but there are also chances of missing many sequences which fall the limit. UCLUST (Edgar, 2010) is the default OTU clustering algorithm in QIIME (version-1.9) (Caporaso et al., 2010) and the most extensively used method. UCLUST is greedy heuristic based method which supports all the OTU electing approaches i.e. closed, *de novo* and open referencing. UCLUST does not sorting the input sequence reads, but it can be customized either to sort by length of the sequences or abundance. At first a sequence read is picked randomly and put as the seed or centroid of the cluster, and then all the other sequence reads are mapped against the seed, if the similarity between seed and target sequence is equal or above some threshold, sequence is included into the cluster which otherwise acts as new seed/centroid. To avoid the computational demand of alignment process UCLUST uses USEARCH. As UCLUST uses USEARCH internally, it is computationally proficient. While comparing with CDHIT, UCLUST can cluster at lower identities and because of the abilities of USEARCH which offers improved sensitivity, UCLUST is able to cluster and classify huge datasets with lower memory usage and high speed (Edgar, 2010). Problem with this type of algorithms is that the result always depends on the order of sequences as every sequence is not compared with each centroid, so a sequence read can be added in the wrong cluster. The older versions of UCLUST were using CDHIT type of similarity, but from version 6 it uses BLAST (Altschul et al., 1990) type identity as the default option. VSEARCH (Rognes et al., 2016) is an open source tool similar to USEARCH. The default identity definition in VSEARCH is based on alignment length without gaps, but it can be customized up to 5 different definitions. VSEARCH can also be customized in sorting strategies for sorting the input. VSEARCH exploits the advantage of Single Instruction Multiple Data stream (SIMD) parallelism and mount up the multithreading in order to accomplish the alignment process at extremely fast move and adopt an optimal global aligner. The two more tools which operates in *de novo* way, SUMACLUST (Mercier et al., 2013) and OTUCLUST (Albanese et al., 2015), exploits the advantage of exact sequence alignment and the clusters are been established in an incremental way by analyzing an ordered list based on abundency of input sequences in counter to previously designated representative set of sequences. SUMACLUST can be used as standalone tool and is also implemented in QIIME pipeline while as OTUCLUST is implemented in MICCA pipeline. SUMACLUST exploits the use of abundance property to sort the given input, implemented with a k-mer filter which is lossless and channeled with a heuristic banded alignment method which is effective for large threshold; finally SIMD architecture is used for parallelism. By default, SUMACLUST bins the given input data with "exact" alignment option. Similar to SUMACLUST, OTUCLUST depends on the perfect sequence alignment to develop the accuracy and to conserve the diversity (Kopylova et al., 2014). K-mer filtering algorithm is used by DNACLUST despite of pairwise alignment method. It sorts input data by length and ensigns un-clustered sequences in twice radius from the centroid so that they cannot be selected up as seeds, but may be put to clusters with an un-flagged cluster center. It will hamper the overlapping of clusters with their centroids within the distance twice of the cluster radius(Ghodsi et al., 2011). Grammar-based distance metric is used by Gram Cluster in order to bin the read sequences. The algorithm creates suffix tree and grammar dictionary for every sequence, then these are being used in sequence similarity for binning of the sequences (Russell et al., 2010). SortMeRNA is a closed reference OTU Clustering metagenomic tool, which is generally used for RNA-Seq data processing in Transcriptomics. It performs pairwise local alignment and is capable of handling large datasets. It performs fragment matching with reference database with sensitivity high and low running time. The query sequence is binned with a sequence having adequate identity and low expectancy value, while the coverage is preserved (Kopylova et al., 2012). In order to optimize the Matthews correlation coefficient (MCC), new OTUs are being reassigned iteratively by OptiClust algorithm (Westcott & Schloss, 2017). The quality of OTU assignments is measure of MCC, which is default metric for valuation & optimization. This method initiates by placing each amplicon read into a single OTU and then by using metrics and iterations which predicts whether the amplicon read should stay in its present OTU or took deviation or in new OTU. DMSC (dynamic multi-seeds clustering) method initially produces clusters/bins based on a threshold in the greedy way. In this method multi-core sequences (MCS) are collected to act as predefined seeds (n-core sequences) in which the threshold distance is always upper than the distance between any sequence read pair. The average of distance to MCS and distance standard deviation metric within MCS is employed to add new sequence reads. The latest allotment to the cluster updates the MCS till no assimilation of sequence to the cluster (Z. G. Wei & Zhang, 2019).

.

C:\Users\Administrator\Desktop\Th-Data\greedy-clustering.tif

**Figure 2 Seed based greedy heuristic clustering based on identity threshold**

**4.2 Hierarchical clustering Approaches**

The hierarchical methods exercise the utilization of pairwise genetic distance matrix, generated by mapping and correlating all reads mutually, and then builds a minimum spanning tree from it called as dendrogram. The clusters are obtained by splitting the dendrogram at some degree decided by the given threshold. The distance between the clusters is measured using graph metrics. The mostly used graph metrics are single, average and complete linkage as shown in Figure 3. Factually each cluster is a sub tree of the complete dendrogram (Olson & F., 1995)(Polanski & Kimmel, 2007). The computational complexity more specifically the time complexity of most of the hierarchical algorithms is quadratic, where as “*n*” is input size (number of sequence reads). The quadratic time complexity for analysis of massive amount of data is a bottleneck and time consuming process in the metagenomic projects. A handful of clustering algorithms have been developed using traditional hierarchical clustering for analyzing biological data, mostly clustering metagenomic sequence reads into OTUs. DOTUR tool is defined as Distance-Based OTU and Richness, this method assigns sequence reads to OTU Clusters for every possible distance using either the furthest, average, or nearest neighbor algorithms. DOTUR then calculates the frequency at which each OTU is pointed out afterwards utilized to check richness estimators, create collectors curves, diversity indices and randomized rarefaction(Schloss & Handelsman, 2005). DOTUR had been later integrated into Mothur in year 2009. Mothur, an open source software pipeline package used for metagenomic data analysis, is frequently used for metagenomic data analysis of NGS sequencing data and holds many algorithmic options (Schloss et al., 2009). ESPRIT was improved to reduce the computational complexity of methods used earlier (Sun et al., 2009). Pairwise globe alignment is used by ESPRIT while multiple sequence alignment tool (MUSCLE) is used by Mothur tool in order to calculate the pairwise distance matrix (Edgar, 2004). One more tool called SLP, is an average neighbor improved version, used to decline the noise and influence of abundant reads to decline OTUs numbers (Huse et al., 2010).

The enhanced version of ESPRIT called as ESPRIT-tree which attains the same accuracy level as that of ESPRIT but has got reduced time complexity (quasi-linear) by employing the use of k-mer filtering approach(Cai & Sun, 2011).The main challenging problem for hierarchical clustering algorithms is the quadratic time and space complexity for big size data. As the hierarchical clustering algorithms need distance matrix of data, hence the flow of steps needs to be sequential and parallelizing is very difficult. For these reasons greedy heuristic algorithms are best candidates for large amount of data.

C:\Users\Administrator\Desktop\Th-Data\linkages.tif

**Figure 3 Linkage methods for hierarchical clustering**

**4.3 Model based Approaches**

Model based methods implement diverse type of mathematical models like Gaussian mixture model, machine learning, Bayesian probability, Fuzzy C-Means, and traditional k-means procedure, etc. There are various model based tools currently active, but in this section some of the important tools of this category are discussed. CROP, an unsupervised Bayesian clustering tool is used to cluster 16S rRNA reads for OTU Prediction, employees Gaussian Mixture Model in getting clusters without using any threshold (Hao et al., 2011). Another Graph based algorithm called M-pick, exploits the graph based concepts, in which the sequences are presumed as vertices of a graph (weighted). An imaginary edge connects the sequence pairs and the weight is represented by the similarity of the sequence pairs. Pairwise sequence alignment is being used to create distance matrix. A connected graph is created from the matrix and then modularity-based community detection algorithm which had been successfully used in OTUs generation in order to capture the structure of community in data (Wang et al., 2013). Swarm v1(Mahé et al., 2014), *de novo* based method, defines an arbitrary global thresholding value by exploiting local threshold value and exploits inside structure of the bin for optimization and abundance for grouping identical sequence reads in an reiterative way. It is a two-step clustering method, which is independent on the order of input sequences and usage of arbitrary global clustering thresholds. The initial step selects sequences in random for the seed and subsequently inspect sequences which differs from the seed by a distance value d and clusters the sequences accordingly. In the next step clusters are divided using abundance. Swarm v2(Mahé et al., 2015), an upgraded version of Swarm v1, has improved time complexity to linear time for large data sizes and also added new fastidious options that minimizes under-grouping of clusters by putting low abundant OTUs into bigger ones. The problem with this method is that it assumes the individual clusters having distances in large between themselves. DMclust is a graph metrics based method. It starts by searching the sequences of dense groups or n-sequence communities, where the threshold is always higher than any distance amidst any b-sequence is not more than a threshold. Then a weighted graph is formed from the dense groups, where the dense groups tends as nodes, every dense group pair is linked by an edge, and the distance of pairwise groups the heaviness of the edge. Then the preclusters are created using a modularity-based community detection algorithm and finally the rest of sequences are added to the closest preclusters to generate OTUs (Z.-G. Wei et al., 2017). IFCM is an enhanced version of fuzzy c-means binning tool, wherein assessed results are being used for dissemination of genome lengths in order to cluster DNA contigs (Liu et al., 2017). In this tool tetra nucleotide frequency is calculated within the contigs followed by cluster number coarsely estimated using genome length distributions from a complete set of non-draft sequenced microbial genomes from NCBI. From the estimated results by the IFCM to bin the DNA contigs and thereafter a function of clustering validity is used to define the results of binning. GeFast extends the generalization of Swarm tools fastidious clustering concept in arbitrary clustering thresholds subsequently adjusting its greediness (Müller & Nebel, 2018). It finishes its working in three phases. First phase initiates by filtering process of input amplicon exploiting the sequence length and alphabet utilizing segment filter and thereafter taking the clustering threshold *t* into account all the amplicon sequences are put into different pools such that each pool receives different amplicon sequences. Then each pool is used separately for clustering and finally the output is generated from the obtained OTUs. SigClust is modified version of k-means algorithm, that maps the sequence reads into binary signatures for clustering the sequence reads (Chappell et al., 2018). The base of the method is encoding of the read sequences as binary signature to model the clustering at an expedient scale. Derivation of binary signatures from k-mers present inside the sequence reads. As SigClust exploits k-means, starting from a set of first cluster centroids with a random then gradually redefining the centroids by replacing the same to the mean of the clusters by relocation of the points into the clusters for which they are close.

1. **Conclusion**

High Throughput Next Generation Sequencing (NGS) spawns gigantic amount of data in extremely little time. It has become a bottleneck because of its large scale size and computational complexity to make use of this data. Clustering algorithms are being used to mine the meaningful insights from this massive data. Similarly Metagenomic clustering algorithms have been created to analyze the NGS data generated from metagenomics projects. In metagenomics, clustering helps in taxonomic profiling using the information of hyper-variable regions of 16S rRNA data. Each algorithm has a different way of working in grouping the metagenomic data. This paper discussed the technical details of metagenomic clustering algorithms and their computational complexities. This would provide a view to the researchers about the current tools in the metagenomic analysis and their usage. Beside it will also provide the elementary knowledge for the enhancement and development of future Metagenomic clustering tools.

**References**

Aggarwal, C. C., & Reddy, C. K. (2014). *Data clustering : algorithms and applications*. CRC Press.

Albanese, D., Fontana, P., De Filippo, C., Cavalieri, D., & Donati, C. (2015). MICCA: A complete and accurate software for taxonomic profiling of metagenomic data. *Scientific Reports*, *5*, 1–7. https://doi.org/10.1038/srep09743

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410. https://doi.org/https://doi.org/10.1016/S0022-2836(05)80360-2

Amelio, A., & Tagarelli, A. (2019). Data Mining: Clustering. In *Encyclopedia of Bioinformatics and Computational Biology* (pp. 437–448). Elsevier. https://doi.org/10.1016/B978-0-12-809633-8.20489-5

Bansal, G., Narta, K., & Teltumbade, M. R. (2018). Next-Generation Sequencing: Technology, Advancements, and Applications. In *Bioinformatics: Sequences, Structures, Phylogeny* (pp. 15–46). Springer Singapore. https://doi.org/10.1007/978-981-13-1562-6\_2

Bernard, G., Pathmanathan, J. S., Lannes, R., Lopez, P., & Bapteste, E. (2018). Microbial Dark Matter Investigations: How Microbial Studies Transform Biological Knowledge and Empirically Sketch a Logic of Scientific Discovery. *Genome Biology and Evolution*, *10*(3), 707–715. https://doi.org/10.1093/gbe/evy031

Bhat, A. H., Nguyen, T. N., Cengiz, K., & Prabhu, P. (2021). An accurate and exact clustering algorithm for next generation sequencing metagenomic sequences. *Mathematical Methods in the Applied Sciences*. https://doi.org/10.1002/MMA.7748

Bhat, A. H., & Prabhu, P. (2017). OTU Clustering : A Window to Analyse Uncultured Microbial World. *International Journal of Scientific Research in Computer Sciences and Engineering*, *5*(6), 62–68. https://doi.org/https://doi.org/10.26438/ijsrcse/v5i6.6268

Bhat, A. H., Prabhu, P., & Balakrishnan, K. (2019). A critical analysis of state-of-the-art metagenomics OTU clustering algorithms. *Journal of Biosciences*, *44*(6), 148. https://doi.org/10.1007/s12038-019-9964-5

Cai, Y., & Sun, Y. (2011). ESPRIT-Tree: Hierarchical clustering analysis of millions of 16S rRNA pyrosequences in quasilinear computational time. *Nucleic Acids Research*, *39*(14), 1–10. https://doi.org/10.1093/nar/gkr349

Caporaso, J. G., Kuczynski, J., Stombaugh, J., Bittinger, K., Bushman, F. D., Costello, E. K., Fierer, N., Peña, A. G., Goodrich, J. K., Gordon, J. I., Huttley, G. A., Kelley, S. T., Knights, D., Koenig, J. E., Ley, R. E., Lozupone, C. A., McDonald, D., Muegge, B. D., Pirrung, M., … Knight, R. (2010). QIIME allows analysis of high-throughput community sequencing data. *Nature Methods*, *7*(5), 335–336. https://doi.org/10.1038/nmeth.f.303

Chappell, T., Geva, S., Hogan, J. M., Huygens, F., Rathnayake, I. U., Rudd, S., Kelly, W., & Perrin, D. (2018). Rapid analysis of metagenomic data using signature-based clustering. *BMC Bioinformatics*, *19*. https://doi.org/10.1186/s12859-018-2540-4

Chen, W., Zhang, C. K., Cheng, Y., Zhang, S., & Zhao, H. (2013). A Comparison of Methods for Clustering 16S rRNA Sequences into OTUs. *PLoS ONE*, *8*(8). https://doi.org/10.1371/journal.pone.0070837

Eckburg, P. B., Bik, E. M., Bernstein, C. N., Purdom, E., Dethlefsen, L., Sargent, M., Gill, S. R., Nelson, K. E., & Relman, D. A. (2005). Diversity of the human intestinal microbial flora. *Science (New York, N.Y.)*, *308*(5728), 1635–1638. https://doi.org/10.1126/science.1110591

Edgar, R. C. (2004). MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, *32*(5), 1792–1797. https://doi.org/10.1093/nar/gkh340

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics*, *26*(19), 2460–2461. https://doi.org/10.1093/bioinformatics/btq461

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S., & Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing*, *2*(3), 267–279. https://doi.org/10.1109/TETC.2014.2330519

Fu, L., Niu, B., Zhu, Z., Wu, S., & Li, W. (2012). CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, *28*(23), 3150–3152. https://doi.org/10.1093/bioinformatics/bts565

Ghodsi, M., Liu, B., & Pop, M. (2011). DNACLUST: Accurate and efficient clustering of phylogenetic marker genes. *BMC Bioinformatics*, *12*(1), 271. https://doi.org/10.1186/1471-2105-12-271

Gilbert, J. A., & Dupont, C. L. (2011). *Microbial Metagenomics : Beyond the Genome*. *November 2010*. https://doi.org/10.1146/annurev-marine-120709-142811

Guinane, C. M., & Cotter, P. D. (2013). Role of the gut microbiota in health and chronic gastrointestinal disease: understanding a hidden metabolic organ. *Therapeutic Advances in Gastroenterology*, *6*(4), 295–308. https://doi.org/10.1177/1756283X13482996

Handelsmanl, J., Rondon, M. R., Goodman, R. M., Brady, S. F., & Clardy, J. (1998). Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products. *Chemistry & Biology*, *5*(10), R245–R249.

Hao, X., Jiang, R., & Chen, T. (2011). Clustering 16S rRNA for OTU prediction: A method of unsupervised Bayesian clustering. *Bioinformatics*, *27*(5), 611–618. https://doi.org/10.1093/bioinformatics/btq725

Hattori, M., & Taylor, T. D. (2009). The human intestinal microbiome: a new frontier of human biology. *DNA Research : An International Journal for Rapid Publication of Reports on Genes and Genomes*, *16*(1), 1–12. https://doi.org/10.1093/dnares/dsn033

Human Microbiome Jumpstart Reference Strains Consortium, T. H. M. J. R. S., Nelson, K. E., Weinstock, G. M., Highlander, S. K., Worley, K. C., Creasy, H. H., Wortman, J. R., Rusch, D. B., Mitreva, M., Sodergren, E., Chinwalla, A. T., Feldgarden, M., Gevers, D., Haas, B. J., Madupu, R., Ward, D. V, Birren, B. W., Gibbs, R. A., Methe, B., … Zhu, D. (2010). A catalog of reference genomes from the human microbiome. *Science (New York, N.Y.)*, *328*(5981), 994–999. https://doi.org/10.1126/science.1183605

Huse, S. M., Welch, D. M., Morrison, H. G., & Sogin, M. L. (2010). Ironing out the wrinkles in the rare biosphere through improved OTU clustering. *Environmental Microbiology*, *12*(7), 1889–1898. https://doi.org/10.1111/j.1462-2920.2010.02193.x

Hussain Bhat, A., & Prabhu, P. (2017). OTU Clustering: A Window to Analyse Uncultured Microbial World. *International Journal of Scientific Research in \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Review Paper. Computer Science and Engineering*, *5*(6), 62–68. www.isroset.org

Janda, J. M., & Abbott, S. L. (2007). 16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: Pluses, perils, and pitfalls. *Journal of Clinical Microbiology*, *45*(9), 2761–2764. https://doi.org/10.1128/JCM.01228-07

Kalpana, B., Prabhu, P., Bhat, A. H., Senthilkumar, A., Arun, R. P., Asokan, S., Gunthe, S. S., & Verma, R. S. (2020). Bacterial diversity and functional analysis of severe early childhood caries and recurrence in India. *Scientific Reports*, *10*(1), 1–15. https://doi.org/10.1038/s41598-020-78057-z

Kopylova, E., Navas-molina, J. A., Mercier, C., & Xu, Z. (2014). Open-source sequence clustering methods improve the state of the art. *MSystems*, *1*(1), 1–16. https://doi.org/http:// doi.org/10.1128/mSystems.00027-16

Kopylova, E., Noé, L., & Touzet, H. (2012). SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics*, *28*(24), 3211–3217. https://doi.org/10.1093/bioinformatics/bts611

Levy, S. E., & Myers, R. M. (2016). Advancements in Next-Generation Sequencing. *Annual Review of Genomics and Human Genetics*, *17*(1), 95–115. https://doi.org/10.1146/annurev-genom-083115-022413

Li, W., & Godzik, A. (2006). Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, *22*(13), 1658–1659. https://doi.org/10.1093/bioinformatics/btl158

Liu, Y., Hou, T., Kang, B., & Liu, F. (2017). Unsupervised Binning of Metagenomic Assembled Contigs Using Improved Fuzzy C-Means Method. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *14*(6), 1459–1467. https://doi.org/10.1109/TCBB.2016.2576452

Lok, C. (2015). Mining the microbial dark matter. *Nature*, *522*(7556), 270–273. https://doi.org/10.1038/522270a

MacCuish, J. D., MacCuish, N. E., & MacCuish, N. E. (2010). *Clustering in Bioinformatics and Drug Discovery* (1st ed.). CRC Press. https://doi.org/10.1201/b10331

Mahé, F., Rognes, T., Quince, C., de Vargas, C., & Dunthorn, M. (2014). Swarm: robust and fast clustering method for amplicon-based studies. *PeerJ*, *2*, e593. https://doi.org/10.7717/peerj.593

Mahé, F., Rognes, T., Quince, C., de Vargas, C., & Dunthorn, M. (2015). Swarm v2: highly-scalable and high-resolution amplicon clustering. *PeerJ*, *3*, e1420. https://doi.org/10.7717/peerj.1420

Marić, J., Križanović, K., Riondet, S., Nagarajan, N., & Šikić, M. (2024). Comparative analysis of metagenomic classifiers for long-read sequencing datasets. *BMC Bioinformatics*, *25*(1). https://doi.org/10.1186/s12859-024-05634-8

Mercier, C., Boyer, F., Bonin, A., & Coissac, E. (2013). SUMATRA and SUMACLUST : fast and exact comparison and clustering of sequences. *Abstract In: SeqBio 25-26th Nov 2013*, 27–29.

Metzker, M. L. (2009). Sequencing technologies — the next generation. *Nature Reviews Genetics*, *11*, 31.

Müller, R., & Nebel, M. E. (2018). GeFaST: An improved method for OTU assignment by generalising Swarm’s fastidious clustering approach. *BMC Bioinformatics*, *19*(1), 1–14. https://doi.org/10.1186/s12859-018-2349-1

Navas-Molina, J. A., Peralta-Sánchez, J. M., González, A., McMurdie, P. J., Vázquez-Baeza, Y., Xu, Z., Ursell, L. K., Lauber, C., Zhou, H., Song, S. J., Huntley, J., Ackermann, G. L., Berg-Lyons, D., Holmes, S., Caporaso, J. G., & Knight, R. (2013). Advancing Our Understanding of the Human Microbiome Using QIIME. In *Methods in enzymology* (Vol. 531, pp. 371–444). https://doi.org/10.1016/B978-0-12-407863-5.00019-8

Nelson, K. E. (Ed.). (2011). *Metagenomics of the Human Body*. Springer New York. https://doi.org/10.1007/978-1-4419-7089-3

Olfa, N. C.-E., & N’Cir, B. (n.d.). *Clustering Methods for Big Data Analytics*. Springer, Cham. https://doi.org/https://doi.org/10.1007/978-3-319-97864-2

Olson, C. F., & F., C. (1995). Parallel algorithms for hierarchical clustering. *Parallel Computing*, *21*(8), 1313–1325. https://doi.org/10.1016/0167-8191(95)00017-I

Pareek, C. S., Smoczynski, R., & Tretyn, A. (2011). Sequencing technologies and genome sequencing. *Journal of Applied Genetics*, *52*(4), 413–435. https://doi.org/10.1007/s13353-011-0057-x

Polanski, Andrzej., & Kimmel, M. (2007). *Bioinformatics* (1st ed.). Springer. https://doi.org/10.1007/978-3-540-69022-1

Rodriguez, M. Z., Comin, C. H., Casanova, D., Bruno, O. M., Amancio, D. R., Costa, L. da F., & Rodrigues, F. A. (2019). Clustering algorithms: A comparative approach. *PLoS ONE*, *14*(1). https://doi.org/10.1371/journal.pone.0210236

Rognes, T., Flouri, T., Nichols, B., Quince, C., & Mahé, F. (2016). VSEARCH: a versatile open source tool for metagenomics. *PeerJ*, *4*, e2584. https://doi.org/10.7717/peerj.2584

Roy, G., Prifti, E., Belda, E., & Zucker, J. D. (2024). Deep learning methods in metagenomics: a review. In *Microbial Genomics* (Vol. 10, Issue 4). Microbiology Society. https://doi.org/10.1099/mgen.0.001231

Russell, D. J., Way, S. F., Benson, A. K., & Sayood, K. (2010). A grammar-based distance metric enables fast and accurate clustering of large sets of 16S sequences. *BMC Bioinformatics*, *11*. https://doi.org/10.1186/1471-2105-11-601

Schloss, P. D., & Handelsman, J. (2005). Introducing DOTUR , a Computer Program for Defining Opera : onal Taxonomic Units and Es : ma : ng Species Richness. *Applied and Environmental Microbiology*, *71*(3), 1501–1506. https://doi.org/10.1128/AEM.71.3.1501

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., Sahl, J. W., Stres, B., Thallinger, G. G., Van Horn, D. J., & Weber, C. F. (2009). Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology*, *75*(23), 7537–7541. https://doi.org/10.1128/AEM.01541-09

Sun, Y., Cai, Y., Liu, L., Yu, F., Farrell, M. L., McKendree, W., & Farmerie, W. (2009). ESPRIT: Estimating Species Richness Using Large Collections of 16S rRNA Shotgun Sequences (Supplementary Data). *Nucl. Acids Res.*, *39*(21), 1–18.

Thompson, L. R., Sanders, J. G., McDonald, D., Amir, A., Ladau, J., Locey, K. J., Prill, R. J., Tripathi, A., Gibbons, S. M., Ackermann, G., Navas-Molina, J. A., Janssen, S., Kopylova, E., Vázquez-Baeza, Y., González, A., Morton, J. T., Mirarab, S., Zech Xu, Z., Jiang, L., … Knight, R. (2017). A communal catalogue reveals Earth’s multiscale microbial diversity. *Nature*, *551*(7681), 457–463. https://doi.org/10.1038/nature24621

Ullman, J.D., & N.d. (n.d.). *Ch. 7 - Clustering* (pp. 240–280). https://doi.org/http://infolab.stanford.edu/~ullman/mmds/ch7.pdf

Wang, X., Yao, J., Sun, Y., & Mai, V. (2013). *M-pick , a modularity-based method for OTU picking of 16S rRNA sequences*. 1–8.

Wei, Z. G., & Zhang, S. W. (2019). DMSC: A dynamic multi-seeds method for clustering 16S rRNA sequences into OTUs. *Frontiers in Microbiology*, *10*(MAR), 1–12. https://doi.org/10.3389/fmicb.2019.00428

Wei, Z.-G., & Zhang, S.-W. (2019). DMSC: A Dynamic Multi-Seeds Method for Clustering 16S rRNA Sequences Into OTUs. *Frontiers in Microbiology*, *10*, 428. https://doi.org/10.3389/fmicb.2019.00428

Wei, Z.-G., Zhang, S.-W., & Zhang, Y.-Z. (2017). DMclust, a Density-based Modularity Method for Accurate OTU Picking of 16S rRNA Sequences. *Molecular Informatics*, *36*(12), 1600059. https://doi.org/10.1002/minf.201600059

Westcott, S. L., & Schloss, P. D. (2015). De novo clustering methods outperform reference-based methods for assigning 16S rRNA gene sequences to operational taxonomic units. *PeerJ*, *3*, e1487. https://doi.org/10.7717/peerj.1487

Westcott, S. L., & Schloss, P. D. (2017). OptiClust, an Improved Method for Assigning Amplicon-Based Sequence Data to Operational Taxonomic Units. *MSphere*, *2*(2), 1–11. https://doi.org/10.1128/mspheredirect.00073-17

Xu, D., & Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, *2*(2), 165–193. https://doi.org/10.1007/s40745-015-0040-1

XU, J. (2006). Microbial ecology in the age of genomics and metagenomics: concepts, tools, and recent advances. *Molecular Ecology*, *15*(7), 1713–1731. https://doi.org/10.1111/j.1365-294X.2006.02882.x