A De Novo Robust Clustering Approach for Amplicon-Based Sequence Data

Alexandre Bazin, Didier Debroas, Engelbert Mephu Nguifo

A De Novo Robust Clustering Approach for Amplicon-Based Sequence Data

Alexandre Bazin[1], Didier Debroas[2], Engelbert Mephu Nguifo[1]

[1]Université Clermont Auvergne, LIMOS, CNRS, Clermont-Ferrand, France
[2]Université Clermont Auvergne, LMGE, CNRS, Clermont-Ferrand, France

Abstract. When analyzing microbial communities, an active and computational challenge concerns the categorization of 16S rRNA gene sequences into operational taxonomic units (OTUs). Established clustering tools use a one pass algorithm in order to tackle high numbers of gene sequences and produce OTUs in reasonable time. However, all of the current tools are based on a crisp clustering approach, where a gene sequence is assigned to one cluster. The weak quality of the output compared to more complex clustering algorithms, forces the user to post-process the obtained OTUs. Providing a membership degree when assigning a gene sequence to an OTU, will help the user during the post-processing task. Moreover it is possible to use this membership degree to automatically evaluate the quality of the obtained OTUs. So the goal of this work is to propose a new clustering approach that takes into account uncertainty when producing OTUs, and improves both the quality and the presentation of the OTUs results.

1 Introduction

Studying the structure of the communities in an ecosystem is central in environmental microbiology [8, 14]. The biosphere's composition can be determined by taking samples in the environment and extracting the DNA sequences through sequencing. From there, these sequences need to be clusterized [4, 9, 11, 15]. As the volume of sequences has drastically increased in recent times, new clustering tools have emerged to treat the data in reasonable time. The algorithms currently used are, from the point of view of algorithmic complexity, the fastest available that do not produce random results. However, due to their simplicity, the results are often of poor quality. These tools being essentially black boxes, their sensitivity to the sequence order, clustering threshold and structure of the data makes it that the users have no way of knowing whether better Operational Taxonomic Units (OTUs) could have been obtained with different parameters or even whether they correctly represent the data. In these circumstances, there is no choice but to blindly trust them.

Distance-based greedy clustering algorithm such as the ones implemented in OTUclust [1], VSEARCH [13], CD-HIT [10] or USEARCH [5] all share the same base algorithm as shown in Algorithm 1.

Algorithm 1: DBG Clustering principle

Input : A set of sequences
Output: A set of OTUs to which the sequences are assigned

```
Clusters = ∅
foreach sequence S do
    foreach known cluster C do
        Compute distance(S, C)
    end
    if a suitable cluster exists then
        Assign S to it
    else
        Create a new cluster with S as the center
    end
end
Return Clusters
```

While more sophisticated algorithms [3, 6, 12, 7, 2] could produce better results quality-wise, their runtime would render them unusable on millions of sequences. As the quality of the OTUs is important, we have to find a way to improve it without increasing the runtime. The different implementations available use a variety of heuristics to counterbalance the simplicity of the algorithm but, to the best of our knowledge, no approach has tried to add a measure of uncertainty to the process. This is why, in order to help increase the quality and trustworthiness of the clustering, we propose to add uncertainty to this simple algorithm through the use of fuzzy clustering.

2 Adding uncertainty to clustering

2.1 Motivation

Distance-based greedy clustering algorithms, such as the one in VSEARCH, produce a number of OTUs and assign each sequence to one of them. The OTU to which a sequence is said to belong to is usually the first one to be encountered that is sufficiently close, i.e. within the specified threshold. This creates two problems :

- A sequence can only belong to a single OTU
- An OTU either includes or does not include a sequence

Having a sequence associated to a single OTU is expected as the ultimate output of the algorithm. For this reason, algorithms usually stop after finding the OTU closest to a sequence, which speeds the computation up. However, not considering all the OTUs a sequence could be assigned to increases the sensitivity to the order - a weakness of these algorithms - and reduces the quality of the clustering.

By using strict thresholds, it is possible to have two nearly identical sequences such that one belongs to a particular OTU while the other does not. This strictness makes it so an OTU partitions the set of sequences into two sets inside of which sequences are considered the same regardless of their distance to the center of the OTU. This lack of distinction between sequences that are isolated and sequences on the border of OTUs hides information that could help understand the data.

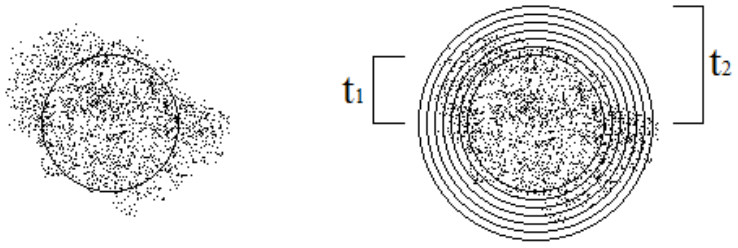
While these would not be problems were the clustering optimal, the need for fast algorithms makes it that the results are not always trustworthy. The OTUs being presented as absolute, the end user has no choice but to consider them correct and cannot know whether the algorithm has encountered ambiguity. We believe that being less strict in the way the OTUs partition sequences would help produce better results from the end user's point of view.

2.2 Fuzzy Clustering

To help increase the quality of the clustering and maximize the information that can be gathered from the data, we propose to add uncertainty to the clustering by means of fuzzy sets.

We define a membership function $f_C(S)$ that, for an OTU C, associates a membership value to a sequence S. Usually, this value is either 0 or 1. Here, we propose to have $f_C(S)$ take its value in $\{\frac{n}{10} \mid n = 0..10\}$. This value represents the degree of membership and, as such, 1 means that the sequence **certainly** belongs to the OTU while 0 means that the sequence **certainly** does not belong to it. Other values represent uncertainty and are used to express that the sequence **nearly** belongs to the OTU. This membership value can easily be computed from the distance between the sequence and the center of the OTU using two thresholds t_1 and t_2 such that $t_1 \geq t_2$. If the distance is lesser than the threshold t_1, the membership value is 1. If the distance is greater than t_2 the value is 0. If the distance is between t_1 and t_2, it increases gradually.

Fig. 1. Representations of a Crisp (Left) and a Fuzzy (Right) Cluster.



Using fuzzy OTUs allows us to discern the difference between sequences close to the OTU and sequences extremely far. Using the parameters t_1 and t_2, we can tune the "detection radius" around OTUs to gather information that would normally be discarded by the clustering algorithm.

3 Evaluating fuzzy OTUs

Having a non-binary membership function produces OTUs that partition the sequences into multiple sets. If we consider only the sequences that belong (more or less) to an OTU, the repartition of their membership values provides information on the topology of the OTU. An ideal OTU would contain only sequences with a membership value of 1, meaning a group of sequences has been perfectly regrouped with a good threshold and no sequence lies ambiguously on the border. More realistically, a good OTU would contain many sequences with high membership values and little sequences with low values. A bad OTU with the majority of its sequences having low membership values could mean that the algorithm has chosen as a center a sequence on the border of a group or, even worse, between two distinct groups.

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
OTU1	6	4	1	1	0	3	8	13	29	88
OTU2	70	41	30	41	34	19	11	6	5	16

We can quickly evaluate the quality of an OTU with this repartition. If we suppose that each sequence lowers the quality of the OTU depending on its membership value, we can use the following formula :

$$Quality(OTU) = 1 - \sum_{i=1}^{9} \omega_i \times \frac{\#\text{ sequences with membership value } i \times 0.1}{\#\text{ sequences in the OTU}}$$

with ω_i being the "cost" of having a sequence with membership value $i \times 0.1$. In our previous examples, and with the following values of ω_i

ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7	ω_8	ω_9
1	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2

we obtain a quality of respectively 0.71 and 0.26 for OTU1 and OTU2, showing OTU1 is better.

A problem arises with singletons that always have perfect quality but these can safely be treated separately.

4 Choosing an OTU

A sequence can belong to multiple OTUs due to fuzzy membership. However, in the end, we want each sequence to be assigned to a single OTU. Hence, we have to choose one of the possible OTUs. We have two types of values left from the clustering process : membership and quality. The first one is based on the distance between the OTU and the sequence and the second one is used to recognize bad OTUs. Choosing the OTU with the best membership value is akin to running VSEARCH while choosing the one with the best quality tends to create bigger OTUs that absorb distant sequences. To better compromise, we can use a linear combination of the two values :

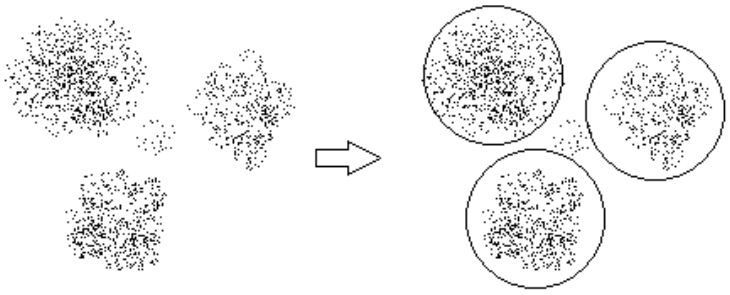
$$\alpha \times quality + \beta \times membership$$

Increasing the importance of the quality reduces the number of OTUs containing sequences. When α is low, the "best" OTUs quality-wise absorb very close sequences that would have been attributed to other OTUs. When α gets too high, the best OTUs start absorbing all the sequences around them, effectively acting like an increase of the distance threshold.

5 Identifying ambiguous sequences

Distance-based greedy algorithms are good at clustering objects that are easy to cluster. Groups of very similar sequences that are different from the rest of the dataset are supposed to birth a new OTU while isolated singletons should be identified to be either removed or treated separately. A problem arises when groups of sequences are close to each other but not enough to be the same OTU. In this case and supposing the algorithm ideally chooses the centers of the OTUs, sequences can lie just between these OTUs. In the current implementations, these ambiguous sequences that must be assigned are usually put in OTUs of their own, increasing the number of OTUs and reducing the overall quality of the clustering.

Fig. 2. A Case of Ambiguous Sequences



Using fuzzy clustering allows us to identify these ambiguous sequences. Using the choice strategy previously mentioned, they can be assigned to a good OTU even though they lie slightly outside of the distance threshold. However, their ambiguousness may be significant for the user. It is thus important to signal their existence and the various fuzzy OTUs they could have alternatively been assigned to.

6 Experimental Results

6.1 Data

We used our algorithm on two datasets with sequences of length between 900 and 2588 for an average of 1156. Separate files contained the real classification of these sequences so that the quality of the clustering could be assessed.

The first dataset contained 3366 sequences and the second one 11028. We used a threshold of 0.97 (97% similarity) for the creation of new OTUs and a threshold of 0.95 for fuzzy membership. The *max_accepts* and *max_rejects* parameters were maxed as to obtain the best result possible. For the choice of the OTU for each sequence, we present the results of two strategies : best quality ($\alpha = 1$ and $\beta = 0$) and compromise ($\alpha = 0.5$ and $\beta = 0.5$).

6.2 Relevant Metrics

To measure the effects of introducing uncertainty to the clustering, we consider the following metrics :

- Computation time in minutes
- Memory usage
- Number of OTUs containing at least a sequence
- Number of OTUs containing a single sequence
- Number of OTUs containing only two sequences
- Average number of sequences per OTU (excluding singletons)
- Number of misclassifications

A misclassification is defined as two sequences being assigned to the same OTU when they should not have been according to the test file.

6.3 Results

First, let us begin with the results on the data set containing 3366 sequences in Table 3.

Then, the data set containing 11028 sequences in Table 4.

Method	Time	Memory	#OTUs	#Singletons	#Pairs	Av. Seq/OTU	#Misclass.
Fuzzy (best quality)	6	358932	1709	1136	292	3.89	922
Fuzzy (compromise)	6	363536	1795	1263	279	3.95	4690
VSEARCH (distance)	6	348608	1864	1365	256	4.01	3658

Fig. 3. Results on 3366 Sequences.

Method	Time	Memory	#OTUs	#Singletons	#Pairs	Av. Seq/OTU	#Misclass.
Fuzzy (best quality)	41	701168	4108	2424	637	5.10	2866
Fuzzy (compromise)	41	696648	4313	2754	614	5.30	10636
VSEARCH (distance)	41	514380	4483	3005	608	5.42	19992

Fig. 4. Results on 11028 Sequences.

6.4 Analysis

Results show that the choice strategy affects every metric relevant to the quality of the clustering : number of OTUs, singletons and pairs, average number of sequences par OTU and number of misclassifications. The fuzzy approach uses more memory than VSEARCH but both choice strategy are similar on this metric. Computation time is identical for all approaches.

We observe that selecting the OTU with the best quality for each sequence produces less OTUs than using the distance. This is due to the fact that some OTUs are initially created centered on isolated sequences near good OTUs. That isolation lowers their quality and the good OTUs absorb their sequences.

The quality approach produces less singletons and more pairs than the distance approach. This most likely means that singletons were created close to either good clusters or one another. The fuzzy approach allows the algorithm to merge those sequences that were slightly too far from the center with their corresponding OTU. The increase in the number of pair appears to be due to the merging of singletons lying too close to one another.

The average number of sequences per OTU decreases when the importance of the quality increases. From this, we deduce that the singletons that disappear are mainly merged with small OTUs.

Finally, the number of misclassifications is greatly reduced by using only the quality for the choice of the OTUs.

7 Discussion

We believe that the experimental results confirm that adding uncertainty to the clustering helps improve the quality of the output. Using fuzzy clusters, we are able to extend the clustering threshold to gather additional information on the OTUs's surroundings and use it to quickly assess their quality. This quality can be used together with the distance to choose an OTU for each sequence. The

resulting output contains less erroneous singletons and misclassifications. Being able to choose the weight of both distance and quality allows for additional tuning.

As previously mentioned, the fuzziness also makes it possible to detect ambiguous sequences and clusters. In our opinion, this is where further work is required. An ambiguous sequence could be arbitrarily assigned to a nearby OTU, become the center of its own OTU or even be considered as an error and deleted but these operations imply such a knowledge of the domain that interactions with the human user become necessary. However, on datasets containing millions of sequences, the number of alerts would render manual treatment impractical or even impossible. Automatizing this treatment would require being able to adapt to the type of data, domain and preferences of the user. We suggest that machine learning techniques be introduced in the process to automatically learn how to handle these ambiguities.

Acknowledgements

This work was supported by the European Union's "*Fonds Européen de Développement Régional (FEDER)*" program and the Auvergne-Rhône-Alpes region.

References

1. Davide Albanese, Paolo Fontana, Carlotta De Filippo, Duccio Cavalieri, and Claudio Donati. Micca: a complete and accurate software for taxonomic profiling of metagenomic data. *Scientific reports*, 5:9743, 2015.
2. Violaine Antoine, Benjamin Quost, Marie-Hélène Masson, and Thierry Denoeux. CECM: constrained evidential c-means algorithm. *Computational Statistics & Data Analysis*, 56(4):894–914, 2012.
3. Violaine Antoine, Benjamin Quost, Marie-Hélène Masson, and Thierry Denoeux. CEVCLUS: evidential clustering with instance-level constraints for relational data. *Soft Comput.*, 18(7):1321–1335, 2014.
4. Wei Chen, Clarence K Zhang, Yongmei Cheng, Shaowu Zhang, and Hongyu Zhao. A comparison of methods for clustering 16s rrna sequences into otus. *PloS one*, 8(8):e70837, 2013.
5. Robert C Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.
6. Isak Gath and Amir B. Geva. Unsupervised optimal fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 11(7):773–780, 1989.
7. Sarra Ben Hariz, Zied Elouedi, and Khaled Mellouli. Clustering approach using belief function theory. In *International Conference on Artificial Intelligence: Methodology, Systems, and Applications*, pages 162–171. Springer, 2006.
8. Mylène Hugoni, Najwa Taib, Didier Debroas, Isabelle Domaizon, Isabelle Jouan Dufournel, Gisèle Bronner, Ian Salter, Hélène Agogué, Isabelle Mary, and Pierre E Galand. Structure of the rare archaeal biosphere and seasonal dynamics of active ecotypes in surface coastal waters. *Proceedings of the National Academy of Sciences*, 110(15):6004–6009, 2013.

9. Weizhong Li, Limin Fu, Beifang Niu, Sitao Wu, and John Wooley. Ultrafast clustering algorithms for metagenomic sequence analysis. *Briefings in bioinformatics*, page bbs035, 2012.
10. Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.
11. Frédéric Mahé, Torbjørn Rognes, Christopher Quince, Colomban de Vargas, and Micah Dunthorn. Swarm: robust and fast clustering method for amplicon-based studies. *PeerJ*, 2:e593, 2014.
12. Airel Pérez-Suárez, José F Martínez-Trinidad, Jesús A Carrasco-Ochoa, and José E Medina-Pagola. Oclustr: A new graph-based algorithm for overlapping clustering. *Neurocomputing*, 121:234–247, 2013.
13. Torbjørn Rognes, Tomáš Flouri, Ben Nichols, Christopher Quince, and Frédéric Mahé. Vsearch: a versatile open source tool for metagenomics. *PeerJ*, 4:e2584, 2016.
14. Simon Roux, Michaël Faubladier, Antoine Mahul, Nils Paulhe, Aurélien Bernard, Didier Debroas, and François Enault. Metavir: a web server dedicated to virome analysis. *Bioinformatics*, 27(21):3074–3075, 2011.
15. Sarah L Westcott and Patrick D Schloss. De novo clustering methods outperform reference-based methods for assigning 16s rrna gene sequences to operational taxonomic units. *PeerJ*, 3:e1487, 2015.