AncestralClust: Clustering of Divergent Nucleotide Sequences by Ancestral Sequence Reconstruction using Phylogenetic Trees

Lenore Pipes [1,*] and Rasmus Nielsen [1,2,3*]

[1] Department of Integrative Biology, University of California-Berkeley, Berkeley, 94707, USA,
[2] Department of Statistics, University of California-Berkeley, Berkeley, CA 94707, USA, and
[3] Globe Institute, University of Copenhagen, 1350 København K, Denmark

*To whom correspondence should be addressed.

Abstract

Motivation: Clustering is a fundamental task in the analysis of nucleotide sequences. Despite the exponential increase in the size of sequence databases of homologous genes, few methods exist to cluster divergent sequences. Traditional clustering methods have mostly focused on optimizing high speed clustering of highly similar sequences. We develop a phylogenetic clustering method which infers ancestral sequences for a set of initial clusters and then uses a greedy algorithm to cluster sequences.
Results: We describe a clustering program *AncestralClust*, which is developed for clustering divergent sequences. We compare this method with other state-of-the-art clustering methods using datasets of homologous sequences from different species. We show that, in divergent datasets, AncestralClust has higher accuracy and more even cluster sizes than current popular methods.
Availability and implementation: AncestralClust is an Open Source program available at https://github.com/lpipes/ancestralclust
Contact: lpipes@berkeley.edu
Supplementary information: Supplementary figures and table are available online.

1 Introduction

Traditional clustering methods such as UCLUST (Edgar, 2010), CD-HIT (Fu *et al.*, 2012), and DNACLUST (Ghodsi *et al.*, 2011) use hierarchical or greedy algorithms that rely on user input of a sequence identity threshold. These methods were developed for high speed clustering of a high quantity of highly similar sequences (Ghodsi *et al.*, 2011; Li *et al.*, 2001; Edgar, 2010) and, generally, these methods are considered unreliable for identity thresholds $<75\%$ because of either the poor quality of alignments at low identities (Zou *et al.*, 2018) or because the performance of the threshold used to count short words drops dramatically with low identities (Huang *et al.*, 2010). At low identities, these methods produce uneven clusters where the majority of sequences are contained in only a few clusters (Chen *et al.*, 2018) and the high variance in cluster sizes reduces the utility of the clustering step for many practical purposes. Clustering of divergent sequences is a fundamental step in genomics analysis because it allows for an early divide-and-conquer strategy that will significantly increase the speed of downstream analyses (Zheng *et al.*, 2018) and clustering of divergent sequences is a frequent request of users of at least one clustering method (Huang *et al.*, 2010). Currently, there are no clustering methods that can accurately cluster large taxonomically divergent metabarcoding reference databases such as the Barcode of Life database (Ratnasingham and Hebert, 2007) in relatively even clusters. Only a few other methods, such as SpClust (Matar *et al.*, 2019) and TreeCluster (Balaban *et al.*, 2019), exist for clustering potentially divergent sequences. SpClust creates clusters based on the use of Laplacian Eigenmaps and the Gaussian Mixture Model based on a similarity matrix calculated on all input sequences. While this approach is highly accurate, the calculation of an all-to-all similarity matrix is a computationally exhaustive step. TreeCluster uses user-specified constraints for splitting a phylogenetic tree into clusters. However, TreeCluster requires an input tree and thus can also be prohibitively slow for large numbers of sequences where a phylogenetic tree is difficult to estimate reliably. With the increasing size of reference databases (Schoch *et al.*, 2020), there is a need for new computationally efficient methods that can cluster divergent sequences. Here we present AncestralClust that was specifically developed for clustering of divergent metabarcoding reference sequences in clusters of relatively even size.

2 Methods

To cluster divergent sequences, we developed AncestralClust which is written in C (Figure 1). Firstly, k random sequences are chosen and the sequences are aligned pairwise using the wavefront algorithm (Marco-Sola *et al.*, 2020). A Jukes-Cantor distance matrix is constructed from the alignments and a neighbor-joining phylogenetic tree is constructed. The Jukes-Cantor model is chosen for computational speed, but more complex models could in principle be used to potentially increase accuracy but also increase computational time. The $C - 1$ longest branches in the tree are then cut to yield C clusters. These subtrees comprise the initial starting clusters. The sequences in each starting cluster are aligned in a multiple sequence alignment using kalign3 (Lassmann, 2020). The ancestral sequences at the root of the tree of each cluster is estimated using the maximum of the posterior probability of each nucleotide using standard programming algorithms from phylogenetics (see e.g., Yang, 2014). The ancestral sequences are used as the representative sequence for each cluster. Next, the rest of the sequences are assigned to each cluster based on the shortest nucleotide distance from the wavefront alignment between the sequence and the C ancestral sequences. If the shortest distance to any of the C ancestral sequences is larger than the average distance between clusters, the sequence is saved for the next iteration. We iterate this process until all sequences are assigned to a cluster. In each iteration after the first iteration, a cut of a branch in the phylogenetic tree is chosen if the the branch is longer that the average length of branches cut in the first iteration. In praxis, only one or two iterations are needed for most data sets if k is defined to be sufficiently large.

We compared AncestralClust to five other state-of-the-art clustering methods: UCLUST (Edgar, 2010), meshclust2 (James and Girgis, 2018), DNACLUST (Ghodsi *et al.*, 2011), CD-HIT (Fu *et al.*, 2012), and SpClust (Matar *et al.*, 2019). We used a variety of measurements to assess the accuracy and evennness of the clustering. We calculated two traditional measures of accuracy, purity and normalized mutual information (NMI), used in Bonder *et al.* (2012). The purity of clusters is calculated as:

$$purity(\Omega, C) = \frac{1}{N} \sum_{k} \max_{j} |\omega_k \cap c_j| \tag{1}$$

where $\Omega = w_1, w_2, ..., w_k$ is the set of clusters, $C = c_1, c_2, ..., c_j$ is the set of taxonomic classes and N is the total number of sequences. NMI is calculated as:

$$NMI(\Omega, C) = \frac{I(\Omega, C)}{[H(\Omega) + H(C)]/2} \tag{2}$$

where mutual information gain is $I(\Omega, C)$ and H is the entropy function. To measure the evenness of the clusters, we used the coefficient of variation which is calculated as:

$$CV = \frac{\sqrt{\sum_i^j (n_i - m)^2/j}}{m} \tag{3}$$

where n_i is the number of sequences in cluster i, j is the total number of clusters, and m is the mean size of the clusters. We also used a taxonomic incompatibility measure to assess the accuracy of the clusters. Let a,b be a pair of species found in cluster i. Incompatibility at a given taxonomic rank is calculated by first identifying the number of times a and b exist in clusters other than cluster i. The total incompatibility is calculated by summing over all pairs of sequences $(a$,$b)$ and all i.

Both NMI and taxonomic incompatibility are very sensitive to the number of clusters and also to unevenness of cluster sizes. To allow fair comparison when numbers of clusters and evenness of cluster sizes vary we, therefore, calculate the *relative NMI* and *relative incompatibility*. These measures are calculated by scaling them relative to their expected values under random assignments given the number of clusters and the cluster sizes. We estimated *relative NMI* by dividing the raw NMI score by the average NMI of 10 clusterings in which sequences have been assigned at random with equal probability to clusters, such that the cluster sizes are same as the cluster sizes produced in the original clustering. The same procedure was used to convert the taxonomic incompatibility measure into relative incompatibility.

3 Results

To first assess performance of clustering methods on divergent nucleotide sequences, we used 100 random samples of 10,000 sequences from three metabarcode reference databases (16, 18S, and Cytochrome Oxidase I (COI)) from the CALeDNA project Meyer *et al.* (2019). We chose to compare our method on this dataset against UCLUST because it is the most widely used clustering program and it performs better than CD-HIT on low identity thresholds (Chen *et al.*, 2018).

We first compared AncestralClust against UCLUST using *relative NMI* and Coefficient of Variation (Figure 2). We used $k = 300$ random initial sequences, which is 3% of the total number of sequences in each sample and $C = 16$ cuts in the initial phylogenetic tree. Notice that the relative NMI tends to be higher with a lower coefficient of variation for AncestralClust across all barcodes. This suggests, that for these divergent eDNA sequences, AncestralClust provides clusterings that are more even in size and that are more consistent with conventional taxonomic assignment. As a second measure of accuracy we measured *relative incompatibility* and coefficient of variation using AncestralClust and UCLUST using for the same datasets under the same running conditions. Notice in Figure 3, AncestralClust tends to create balanced clusters with lower relative taxonomic incompatibilities compared to UCLUST at all taxonomic levels. Similar results are seen for metabarcode 18S (Fig S1). However, for metabarcode 16S (Fig S2), AncestralClust performs noticeably better than UCLUST at the species, genus, and family levels but at the order, class, and phylum levels it performs either the same or worse. Also, at the species, genus, and family levels, it is apparent that as the UCLUST clusters approach a lower coefficient of variation, the *relative incompatibility* increases dramatically.

Next, we analyzed two datasets with different properties: one dataset of diverse species from the same gene and another dataset of homologous genes from species of the same phyla. In the first dataset, we expect that the sequences to cluster according to species. In the second dataset, we expect the sequences to cluster according to different genes. We compared AncestralClust to four commonly used clustering programs (UCLUST, meshclust2, CD-HIT2, and DNACLUST) and one clustering program designed for divergent sequences, SpClust. The first dataset contained 13,043 sequences from the COI CaleDNA database from 11 divergent species that were from 7 different phyla and 11 different classes and the second data set contained sequences from 6 different genes from taxonomically similar species. First, we compared all methods using 13,043 COI sequences from the 11 different species (Table 1). We expect these sequences to form 11 different clusters, each including all the sequences from one species. We chose identity thresholds to enforce the expected number of clusters for each method. We were unable to form 11 clusters using CD-HIT because the program does not allow clustering of sequences with identity thresholds $< 80\%$ at default parameters. For SpClust, we used the three precision modes available for the method. In this analysis, AncestralClust achieved a perfect clustering (the purity was 1 and *relative incompatibility* was 0) although it was the second slowest, and had the second lowest memory requirements. UCLUST was one of the fastest methods and used the least amount of memory but had the second lowest purity with third highest *relative NMI* values. meshclust2 had no incompatibilities and the second highest purity and relative NMI values but was the third slowest method. DNACLUST had the most uneven clusters and the second lowest *relative NMI* value with the highest *relative incompatibility*. SpClust only identified one cluster, with a computational time of ~2 days. In comparison, AncestralClust took ~5 minutes and UCLUST used $<$ 1 second.

Next, we analyzed 'genomic set 1' from Matar *et al.* (2019), which consists of 39 sequences from 6 homologous genes (FCER1G, S100A1, S100A6, S100A8, S100A12, and SH3BGRL3 in Table 2). We expect these sequences to form 6 clusters. We varied the identity thresholds for UCLUST and meshclust2 using thresholds 0.4, 0.6, and 0.8. For CD-HIT, we used the lowest identity threshold available on default parameters which is 0.8. We were unable to use DNACLUST for this analysis because it cannot handle sequences longer than 4500bp (the average sequence length was 2,387.9bp and the longest sequence was 5,363bp). Since this dataset contained 6 different genes, we calculated *relative NMI* using genes as the classes and did not use incompatibility as an accuracy measure. Only AncestralClust, UCLUST, and meshclust2 produced the expected number of clusters, and among the methods that created the expected number of clusters, AncestralClust had the highest purity value. AncestralClust was the second slowest method and had the highest memory requirements which is due to the wavefront algorithm alignment which is $\mathcal{O}(s^2)$ in memory requirements where s is the alignment score. Since alignments were performed using 6 different genes that were longer than 1.5kb, this resulted in a high value of s. SpClust had the highest *relative NMI* using all precision modes and the same purity as AncestralClust for its moderate and maximum precision modes, however, failed to produce the expected number of clusters.

4 Conclusions

We developed a phylogenetic-based clustering method, AncestralClust, specifically to cluster divergent metabarcode sequences. We performed a comparative study between AncestralClust and widely used clustering programs such as UCLUST, CD-HIT, DNACLUST, meshclust2, and for divergent sequences, SpClust. UCLUST and DNACLUST are substantially faster than AncestralClust and should be the preferred method if computational speed is the main concern. However, AncestralClust tends to form clusters of more even size with lower taxonomic incompatibility and higher NMI than other methods, for the relatively divergent sequences analyzed here. We recommend the use of AncestralClust when sequences are divergent, especially if a relatively even clustering is also desirable, for example for various divide-and-conquer approaches where computational speed of downstream analyses increases faster than linearly with cluster size.

Acknowledgements

This work used the Extreme Science and Engineering Discovery Environment (XSEDE) Bridges system at the Pittsburgh Supercomputing Center through allocation BIO180028.

References

Balaban, M., Moshiri, N., Mai, U., Jia, X., and Mirarab, S. (2019). Treecluster: Clustering biological sequences using phylogenetic trees. *PloS one*, **14**(8), e0221068.

Bonder, M. J., Abeln, S., Zaura, E., and Brandt, B. W. (2012). Comparing clustering and pre-processing in taxonomy analysis. *Bioinformatics*, **28**(22), 2891–2897.

Chen, Q., Wan, Y., Zhang, X., Lei, Y., Zobel, J., and Verspoor, K. (2018). Comparative analysis of sequence clustering methods for deduplication of biological databases. *J. Data and Information Quality*, **9**(3).

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than blast. *Bioinformatics*, **26**(19), 2460–2461.

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics*, **28**(23), 3150–3152.

Ghodsi, M., Liu, B., and Pop, M. (2011). Dnaclust: accurate and efficient clustering of phylogenetic marker genes. *BMC bioinformatics*, **12**(1), 1–11.

Huang, Y., Niu, B., Gao, Y., Fu, L., and Li, W. (2010). CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, **26**(5), 680–682.

James, B. T. and Girgis, H. Z. (2018). Meshclust2: Application of alignment-free identity scores in clustering long dna sequences. *bioRxiv*, page 451278.

Lassmann, T. (2020). Kalign 3: multiple sequence alignment of large datasets.

Li, W., Jaroszewski, L., and Godzik, A. (2001). Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, **17**(3), 282–283.

Marco-Sola, S., Moure López, J. C., Moreto Planas, M., and Espinosa Morales, A. (2020). Fast gap-affine pairwise alignment using the wavefront algorithm. *Bioinformatics*, (btaa777), 1–8.

Matar, J., Khoury, H. E., Charr, J.-C., Guyeux, C., and Chrétien, S. (2019). Spclust: Towards a fast and reliable clustering for potentially divergent biological sequences. *Computers in biology and medicine*, **114**, 103439.

Meyer, R. S., Curd, E. E., Schweizer, T., Gold, Z., Ramos, D. R., Shirazi, S., Kandlikar, G., Kwan, W.-Y., Lin, M., Freise, A., *et al.* (2019). The california environmental dna "caledna" program. *bioRxiv*, page 503383.

Ratnasingham, S. and Hebert, P. D. (2007). Bold: The barcode of life data system (http://www. barcodinglife. org). *Molecular ecology notes*, **7**(3), 355–364.

Schoch, C. L., Ciufo, S., Domrachev, M., Hotton, C. L., Kannan, S., Khovanskaya, R., Leipe, D., Mcveigh, R., O'Neill, K., Robbertse, B., *et al.* (2020). Ncbi taxonomy: a comprehensive update on curation, resources and tools. *Database*, **2020**.

Yang, Z. (2014). *Molecular evolution: a statistical approach*. Oxford University Press.

Zheng, W., Mao, Q., Genco, R. J., Wactawski-Wende, J., Buck, M., Cai, Y., and Sun, Y. (2018). A parallel computational framework for ultra-large-scale sequence clustering analysis. *Bioinformatics*, **35**(3), 380–388.

Zou, Q., Lin, G., Jiang, X., Liu, X., and Zeng, X. (2018). Sequence clustering in bioinformatics: an empirical study. *Briefings in Bioinformatics*, **21**(1), 1–10.

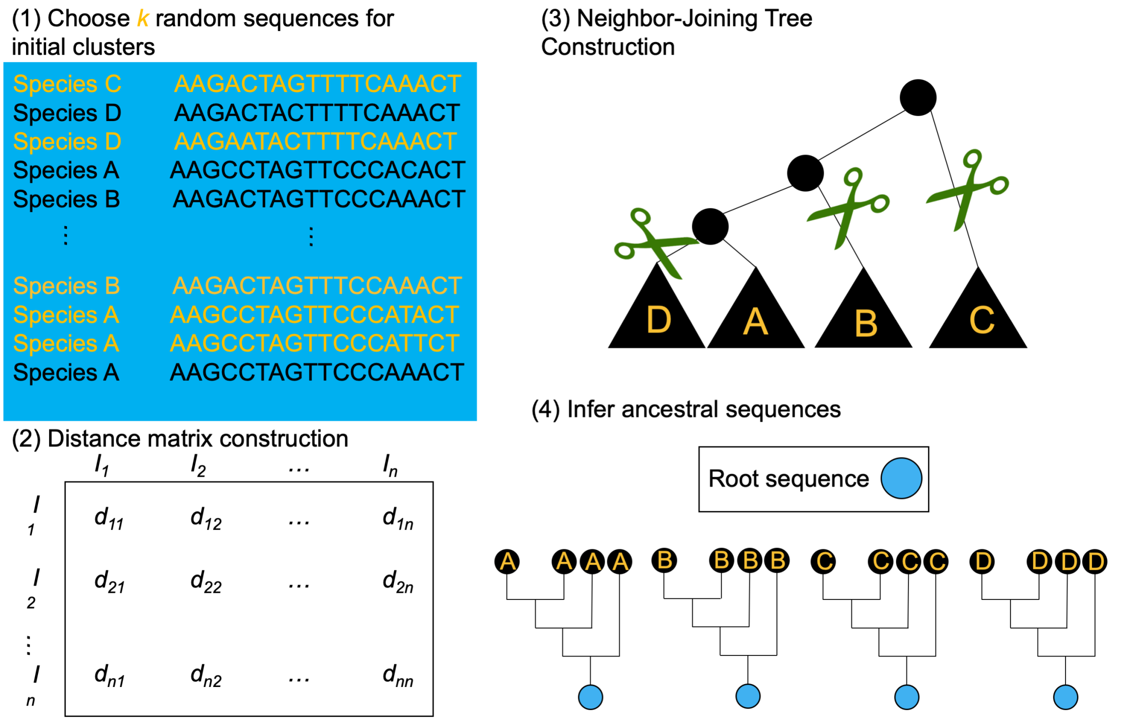
(1) Choose k random sequences for initial clusters
Species C AAGACTAGTTTTCAAACT
Species D AAGACTACTTTTCAAACT
Species D AAGAATACTTTTCAAACT
Species A AAGCCTAGTTCCCACACT
Species B AAGACTAGTTCCCAAACT
Species B AAGACTAGTTTCCAAACT
Species A AAGCCTAGTTCCCATACT
Species A AAGCCTAGTTCCCATTCT
Species A AAGCCTAGTTCCCAAACT
(2) Distance matrix construction
l_1 l_2 … l_n
d_{11} d_{12} … d_{1n}
d_{21} d_{22} … d_{2n}
d_{n1} d_{n2} … d_{nn}
(3) Neighbor-Joining Tree Construction
D A B C
(4) Infer ancestral sequences
Root sequence
A A A A B B B B C C C C D D D D

Figure 1. Overview of AncestralClust. In (1), k random sequences are chosen for the initial clusters. (2) Using the k sequences a distance matrix is constructed. Using the distance matrix, a neighbor-joining tree is constructed and $C - 1$ cuts are made to create C clusters. In (4), each cluster is multiple sequenced aligned and the ancestral sequences are reconstructed in the root node of each tree. The rest of the unassigned sequences are then aligned to the ancestral sequences of each cluster and the shortest distance to each ancestral sequence is calculated. The process is iterated until all sequences are assigned to a cluster.

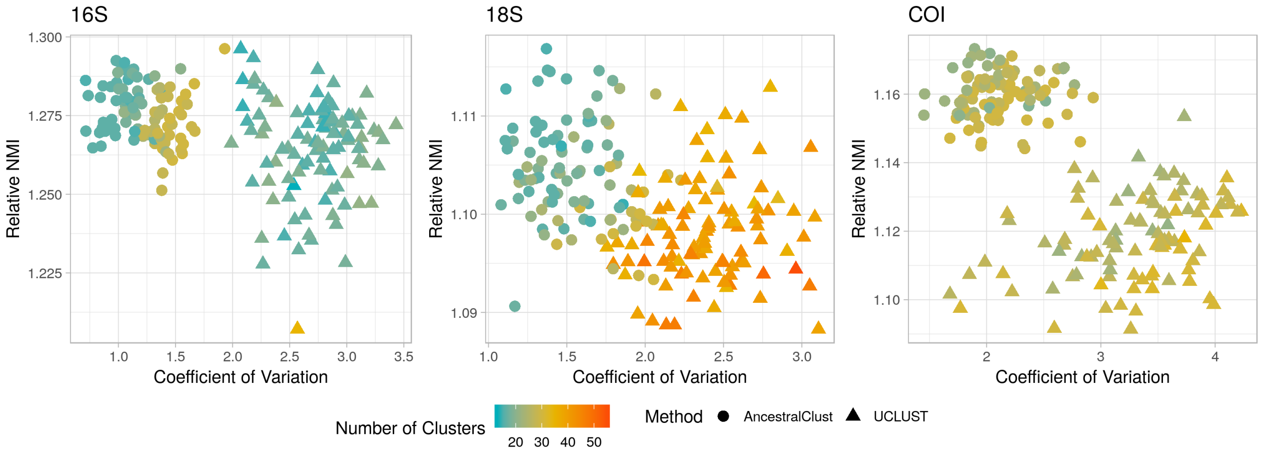

Figure 2. Relative NMI against coefficient of variation for AncestralClust and UCLUST for 100 samples of 10,000 randomly chosen 16S, 18S, and COI reference sequences from the CALeDNA Project (Meyer et al., 2019). The similarity threshold for UCLUST was 0.58. For AncestralClust, we used 300 initial random sequences with 15 initial clusters. Relative NMI was calculated by dividing NMI by the average of 10 random samples of the same fixed cluster size.

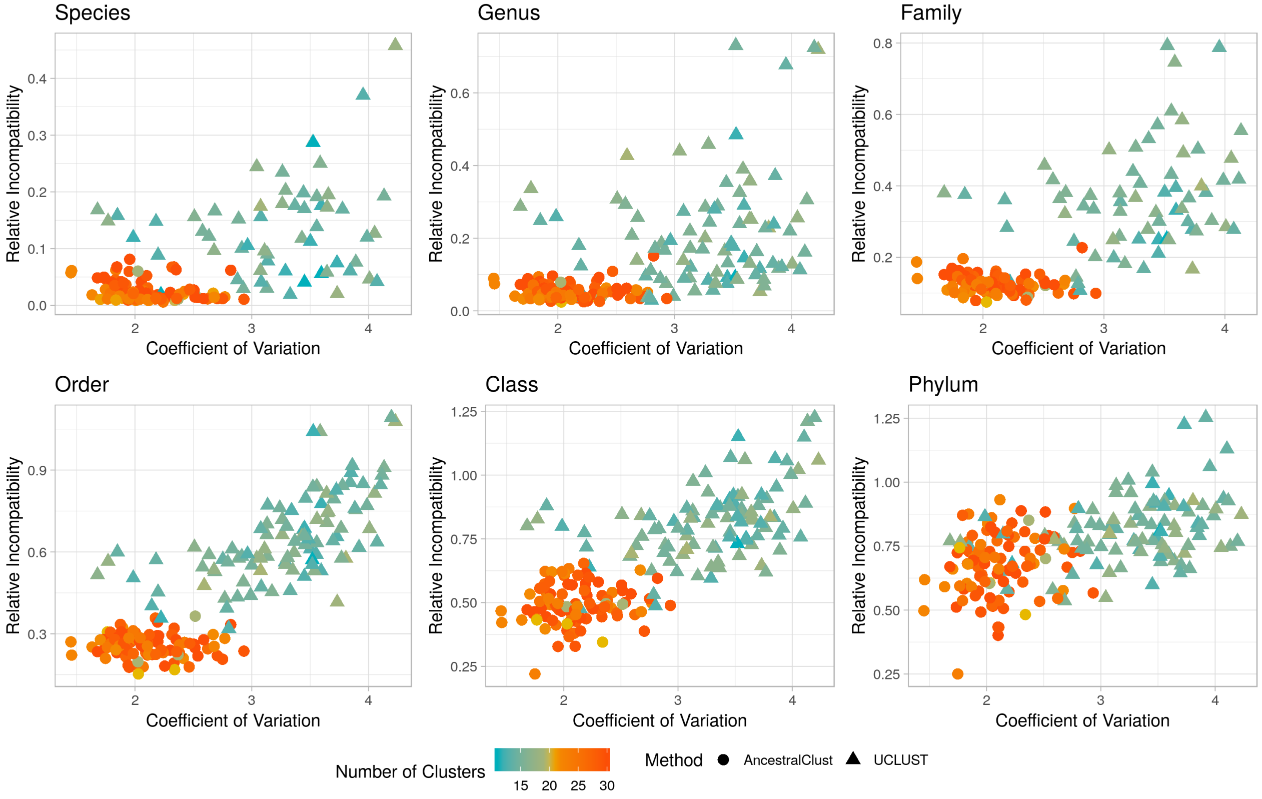

Figure 3. Relative incompatibility against coefficient of variation for AncestralClust and UCLUST for 100 samples of 10,000 randomly chosen COI reference sequences. COI reference sequences are from the CALeDNA Project (Meyer et al., 2019). The similarity threshold for UCLUST was 0.58. For AncestralClust, we used 300 initial random sequences with 15 initial clusters.

Table 1. Comparisons of clustering methods using 13,043 COI sequences from 11 different species. The list of species can be found in Table S1. Incompatibility was calculated at the taxonomic rank of species. For UCLUST, meshclust2, and DNACLUST, the identity thresholds were chosen to force the expected 11 number of clusters. For CD-HIT, the lowest possible identity was chosen which is 0.8. In the case of SpClust, Coefficient of Variation cannot be calculated for 1 cluster. SpClust clusters were created with version 2.

Method	# of clusters	Time (sec)	Mem (MB)	Purity	Relative Incompat. (species)	Relative NMI	Coeff. of Var.
AncestralClust	11	293.2	19.3	1	0	551.09	0.8574
UCLUST	11	<1	9.9	0.8717	0.0182	474.63	0.8300
meshclust2	11	108.14	46.5	0.9855	0	498.898	0.1053
CD-HIT	24	5.86	43.9	0.8561	0	241.66	1.2031
DNACLUST	11	<1	170.6	0.9455	0.0545	24.21	1.8987
SpClust (fast)	1	152046.5	2678.9	1	0	1	-
SpClust (moderate)	1	188172.9	6457.6	1	0	1	-
SpClust (maxPrecision)	1	189577.1	6452.5	1	0	1	-

Table 2. Comparisons of clustering methods using 39 sequences from 6 homologous genes from Matar et al. (2019).'id' refers to the identity threshold used. We used identity thresholds of 0.4, 0.6, and 0.8 for UCLUST and meshclust2. We used precision levels of fast, moderate, and maximum for SpClust using version 1 since version 2 only produced 1 cluster for all modes. DNACLUST has a maximum sequence length of 4500bp and could not be used on this dataset.

Method	**# of clusters**	**Time (sec)**	**Memory (Mb)**	**Purity**	**Relative NMI**	**Coefficient of Variation**
AncestralClust	6	370.3	412.0	0.9487	1.8660	0.3982
UCLUST (id=0.4)	6	1	15.4	0.7436	1.5667	0.5396
UCLUST (id=0.6)	19	1	20.1	0.7179	1.4379	0.7166
UCLUST (id=0.8)	29	1.9	20.4	0.5641	1.1717	0.4565
meshclust2 (id=0.4)	6	1.1	7.7	0.8462	1.6625	1.2489
meshclust2 (id=0.6)	10	2.9	8.8	0.7949	1.9257	1.071
meshclust2 (id=0.8)	26	2.4	9.4	0.6410	1.2240	0.6325
SpClust (fast)	4	44.6	166.2	0.8718	2.2463	0.8432
SpClust (moderate)	4	112.5	166.1	0.9487	2.4335	0.6453
SpClust (max precision)	4	570.1	166.0	0.9487	2.9449	0.6809
CD-HIT (id=0.8)	31	0.48	39.9	0.4103	1.0950	0.4574
DNACLUST	-	-	-	-	-	-