BMC Bioinformatics

RESEARCH

Open Access

Clustering biological sequences with dynamic sequence similarity threshold

Jimmy Ka Ho Chiu and Rick Twee-Hee Ong*

*Correspondence:
ephoth@nus.edu.sg
Saw Swee Hock School of Public Health, National University of Singapore and National University Health System, Singapore 117549, Singapore

Abstract

Background: Biological sequence clustering is a complicated data clustering problem owing to the high computation costs incurred for pairwise sequence distance calculations through sequence alignments, as well as difficulties in determining parameters for deriving robust clusters. While current approaches are successful in reducing the number of sequence alignments performed, the generated clusters are based on a single sequence identity threshold applied to every cluster. Poor choices of this identity threshold would thus lead to low quality clusters. There is however little support provided to users in selecting thresholds that are well matched with the input sequences.

Results: We present a novel sequence clustering approach called ALFATClust that exploits rapid pairwise alignment-free sequence distance calculations and community detection in graph for clusters generation. Instead of a single threshold applied to every generated cluster, ALFATClust is capable of dynamically determining the cut-off threshold for each individual cluster by considering both cluster separation and intra-cluster sequence similarity. Benchmarking analysis shows that ALFATClust generally outperforms existing approaches by simultaneously maintaining cluster robustness and substantial cluster separation for the benchmark datasets. The software also provides an evaluation report for verifying the quality of the non-singleton clusters obtained.

Conclusions: ALFATClust is able to generate sequence clusters having high intra-cluster sequence similarity and substantial separation between clusters without having users to decide precise similarity cut-off thresholds.

Keywords: Sequence clustering, Graph clustering, Homologous sequences, Metagenomics

Background

Sequence clustering refers to the process of grouping together similar biological sequences such that only homologous sequences are expected to appear in each sequence cluster. It is particularly useful for identifying various sets of potentially homologous candidates from unknown sequences for further analysis or annotation, as well as aggregating sequencing reads for reference genes abundance estimation in metagenomic samples. Sequence clustering could be considered within the problem domain of general data clustering and are usually resolved using unsupervised learning techniques. Early approaches exploit agglomerative hierarchical clustering [1] to cluster sequences with either single linkage (e.g. BlastClust [2] and GeneRAGE [3]) or average linkage (methods proposed by Loewenstein et al. [4] and Uchiyama [5]) metrics. Partitional clustering, especially K-means clustering [6, 7], was another popular method used to derive sequence clusters (customized K-means approaches by Ashlock et al. [8] and Kelarev et al. [9]). All these approaches require a pairwise sequence distance matrix to be computed. High computation costs are therefore incurred due to the O(N^2) pairwise sequence alignments (such as BLAST alignments [10]) required for N sequences. The size of the distance matrix ($N \times N$) also creates a scalability problem in terms of space complexity. Moreover, clustering results are often sensitive to user-specified clustering parameters. For example, the K-means algorithm would require the number of final clusters K to be specified upfront. In order to determine the optimal parameter values for any given set of sequences, the clustering process would need to be performed iteratively, each with a different set of parameter values. After each iteration, an internal validation index [11] such as the silhouette coefficient [12] is calculated from the pairwise sequence distance matrix for the generated output clusters. The set of parameter values having the best index score is deemed to generate the optimal clusters. However, the number of sequences in each cluster is expected to vary substantially [13], with some clusters having hundreds of sequences while some having only a few or even just a single sequence, therefore making it difficult to efficiently estimate the clustering parameters by assuming similar size or density for all clusters.

Many recent sequence clustering approaches [14–18] therefore aim at minimizing both the number of pairwise alignments performed and the space complexity (i.e. the sequence distance matrix is no longer required), as well as defining the cluster cut-off through a biologically comprehensible parameter known as the sequence identity threshold T. The value of T ranges from 0 (complete mismatch) to 1 (identical sequences) and is usually selected based on users' domain knowledge or a widely accepted value within the domain. It thus reflects that the higher value of the threshold, the more similar would be the sequences in each derived cluster. Such approaches are primarily greedy algorithms that assign a target sequence to an existing cluster when the pairwise sequence identity of this target sequence with the representative center sequence of this cluster is at least T, or otherwise creates a new cluster of size = 1 with this target sequence as its representative center sequence. Techniques such as short word filtering [19, 20] can be applied to avoid the computational expensive procedure of aligning a target sequence with the representative center sequence if their pairwise sequence identity is likely below T. The greedy algorithms are also very space efficient because they only consider the sequence identities between the target sequence and all existing representative center sequences. The clustering results can however be affected by the choices of the representative center sequences for the clusters, which are determined by the order of the sequences assessed. MeShClust [21] therefore addresses this limitation by reselecting the representative center sequence from all sequences in the cluster after a new sequence has been added using the mean-shift algorithm [22]. The representative center sequences for all clusters are also reselected again by mean-shift once all sequences have been clustered, and two clusters are merged when the pairwise sequence identity between their representative center sequences is equal to or greater than T.

In contrast, MMseqs2 [23] models each sequence as a unique graph vertex, and two vertices are connected by an edge when the pairwise alignment of their underlying sequences satisfies particular criteria including significance, sequence coverage, and T. Sequence clusters are then obtained through a graph clustering approach. In addition, the sequence clustering tool Linclust [24] can be run as a pre-processing step to divide the sequences into intermediate clusters for individual graph clustering in each intermediate cluster for scalability. For computational efficiencies, MMseqs2 replaces the exact alignment process between sequences with rapid approximations. The speedup techniques utilized by the different algorithms are summarized in Additional file 1: Table S1. Although the sequence identity threshold T is comprehensible to most users, a poorly chosen value could generate clusters substantially different from the true biological clusters [21]. The most common scenario is that the value of T is set too high and hence some homologous sequences are assigned to different clusters, but these clusters are hardly identifiable given the large number of sequences or insufficient annotation. Parameter optimization with internal validation index is also not feasible for these approaches due to the absence of the complete distance matrix.

We therefore develop a novel sequence clustering method that dynamically adjusts the cut-off thresholds for individual clusters. It first estimates a complete pairwise sequence distance matrix using an alignment-free approach to avoid performing the traditional slower sequence alignments. This distance matrix is then used to derive all the edge weights for a graph, in which each sequence is represented by a vertex, and the edge weight for an edge denotes the pairwise similarity between the pair of sequences associated with. Sequence clustering is now performed via an iterative graph clustering in which each vertex is regarded as a singleton graph cluster (a singleton graph cluster consists of only one vertex) initially. Each iteration begins with identifying potential clusters to be merged. A cluster separation cut-off threshold then determines which of them can be merged into a single cluster without introducing possible outliers. When a cluster is prohibited from expansion, it is deemed well separated from its neighbours. In contrast to the sequence identity threshold T, the cluster separation cut-off threshold is a dynamic threshold because it is partially determined by the clusters considered. We thus name this sequence clustering approach as Alignment-Free Adaptive Threshold Clustering, or ALFATClust in short. ALFATClust is implemented as a publicly available tool, which also provides an user option to evaluate the non-singleton clusters in terms of sequence identity through sequence alignment.

The remaining sections of this manuscript are organized as follows: The "Methods" section first introduces the sequence distance calculation approach and its estimation parameters to be used in ALFATClust, and then gives an overview of the ALFATClust algorithm. Its core steps such as the binning process to derive the graph clusters and the graph contraction are illustrated afterwards. Details of the scalability enhancement and the optional sequence cluster evaluation report are also provided. In the "Results" section, we assess both clustering and time performances of ALFATClust with other sequence clustering tools using the benchmark datasets. We then elaborate on the advantages and limitations of ALFATClust in the "Discussion" section.

Methods

Pairwise sequence distance using alignment-free approach

Mash [25] and Dashing [26] are two rapid alignment-free sequence distance approaches that require k-mer size and sketch size to be specified as input parameters. We therefore performed experiments to assess the distance calculation method to be implemented in ALFATClust and also determine the relevant input parameters. The experimental details can be found in Additional file 1 provided. Results (Additional file 1: Figures S1–S5) indicate that Mash is better due to the higher correlation between Mash distance and the sequence distance obtained by alignment approach. In addition, from Additional file 1: Figures S1 and S3, the optimal Mash k-mer size for gene sequences is 17, or lower (e.g. 13) when some sequences are very short. For complete protein sequences, the optimal k-mer size is 9 (Additional file 1: Figure S5). The optimal sketch size is 2000 for both gene and protein sequences. Since Mash distance d ranges from 0 to 1, the corresponding sequence similarity can be easily calculated as $1-d$.

Overview of ALFATClust algorithm

The ALFATClust algorithm consists of four components:

1. Mash [25] distance calculation for constructing a graph to model pairwise distances between the input sequences;
2. Leiden algorithm [27] to partition the graph into communities [28] of raw graph clusters;
3. A binning process to bin the vertices in each raw graph cluster into validated graph clusters;
4. A graph contraction step to replace every validated graph cluster by a single vertex.

Suppose a Mash distance matrix D is computed for a sequence set S such that d_{ij} of D represents the pairwise Mash distance between sequences s_i and s_j in S. Since Mash distance is a symmetric distance measure, $d_{ij}=d_{ji}$ and both D and $W=1-D$ are therefore symmetric matrices. An undirected weighted graph $G=(V, E, W)$ is initialized with each vertex $v_i \in V$ representing a sequence $s_i \in S$. Every vertex pair (v_i, v_j) $(i<j)$ is connected with an edge $e_{ij} \in E=\{(v_i, v_j) \mid v_i, v_j \in V, i<j\}$, and the edge weight of e_{ij} is equal to w_{ij} of W. $w_{ii}=1$ despite the absence of self-loop e_{ii} in G. w_{ij} therefore denotes the sequence similarity between sequences s_i and s_j. Leiden algorithm then partitions the graph into raw graph clusters (communities) that maximize the score calculated using a pre-defined quality function. Given a value of γ between 0 and 1, the quality function of the Constant Potts Model (CPM) [29] guarantees that for any vertex v in a non-singleton raw graph cluster R, the average edge weight of edges connecting between v and other vertices in R exceeds γ. This means the average sequence similarity between the sequences within every non-singleton raw cluster is higher than γ. The core parameters of the ALFATClust algorithm thus include a value range $[\gamma_{low}, \gamma_{high}]$ $(\gamma_{low}<\gamma_{high})$ for γ and a step size Δ, which are used to determine the values of γ used in the algorithm:

1. Create graph G using Mash distance matrix D; assign γ_{high} to γ

```
2. While $\gamma > \gamma_{low} - \Delta$ do:
    3. Run Leiden algorithm on $G$ using CPM with $\gamma$ to obtain set of raw clusters $C_{raw}$
    4. Initialize $L$ with an empty list
    5. Assign $C_{raw}$ to $L$ if $\gamma = \gamma_{high}$; otherwise bin each raw cluster in $C_{raw}$ and add the bins to $L$
    6. Contract $G$ by replacing every graph cluster in $L$ by a single vertex and update edges
    7. Decrement $\gamma$ by $\Delta$
8. Return the sequence clusters represented by $L$ derived in the final iteration
```

The above-mentioned steps would derive graph (sequence) clusters iteratively from $\gamma = \gamma_{high}$ to a value determined by γ_{low}. In the i-th iteration, it first performs Leiden algorithm using CPM with $\gamma = \gamma_{high} - (i-1)\Delta$, and the raw clusters C_{raw} become the graph clusters L directly in the first iteration. From the second iteration onwards, the vertices in each individual raw cluster are binned (refer to the binning section below) and vertices in the same bin become a validated graph cluster in L. After deriving L for the current iteration, graph G is contracted in a way similar to Uchiyama's approach [5] that all vertices belonging to a cluster in L are replaced by a single vertex. The next iteration, if any, begins with the contracted graph G. The graph clusters L derived in the final iteration thus become the output sequence clusters. Details of the Leiden algorithm with CPM, binning, and graph contraction are further elaborated below. A detailed description of the ALFATClust algorithm can be found in Additional file 1.

Leiden algorithm with Constant Potts Model to identify vertices for binning

Community detection [28] refers to the process of partitioning a graph such that vertices in each partition are closely related with each other. While each of these partitions is originally known as a community, we term it as a raw graph cluster or raw cluster in this article. In ALFATClust, community detection is performed using the Leiden algorithm [27] which is an improvement of the Louvain algorithm [30] in both cluster quality and execution time. Briefly, Leiden algorithm starts with each vertex as an individual cluster, and then updates the clusters iteratively to maximize the overall quality score q. The calculation of q requires a quality function to quantify how closely related the vertices are within each of the raw clusters. For a graph G without self-loop, Eq. (1) is the general form of the quality function as defined in CPM [29]:

$$q = \sum_{i<j} \left(\alpha_{ij} w_{ij} - \gamma\right) \delta\left(\sigma_i, \sigma_j\right) \tag{1}$$

where α_{ij} belongs to the adjacency matrix A. $\alpha_{ij} = 1$ for any i and j because G is complete (i.e. an edge exists between any two distinct vertices regardless of edge weight). σ_i refers to the raw cluster for v_i, and $\delta(\sigma_i, \sigma_j) = 1$ if $\sigma_i = \sigma_j$, i.e., both v_i and v_j belong to the same raw cluster, and zero otherwise. CPM therefore only considers edge weights for edges within a cluster but not those across clusters. The value of γ, which is called the resolution parameter, is within the range of the edge weight for G, i.e. from 0 to 1. When a vertex exists as a singleton raw cluster it contributes zero score to q according to Eq. (1). It follows that the baseline case for CPM is every raw cluster being a singleton cluster and therefore $q=0$ regardless of the value of γ. Given a value of γ, for any vertex v in a non-singleton raw cluster R, the average edge weight of the intra-cluster edges in R involving v must be greater than γ in order to contribute a positive score to q. γ therefore becomes a lower bound of the average intra-cluster edge weight for all raw clusters. Graph clusters maximizing the value of q are regarded as the raw clusters for a particular value of γ.

A problem for community detection is the size bias where large communities dominate over small communities [31]. Similar size bias are also observed in CPM where the lower the value of γ, the more it favours larger raw cluster size over higher edge weight. For example, suppose graph G consists of vertices v_1, v_2, v_3, v_4, v_5, … with edge weights w_{ij} in W. For $\gamma=0.875$, v_1 may form a raw cluster $R=\{v_1, v_2\}$ when $w_{12}=0.89$, thus giving $q=0.89-0.875=0.15$. Meanwhile, v_1 cannot form a larger raw cluster $R'=\{v_1, v_3, v_4, v_5\}$ when $\{w_{13}, w_{14}, w_{15}, w_{34}, w_{35}, w_{45}\}=\{0.84, 0.86, 0.87, 0.85, 0.87, 0.86\}$, because q for R' is equal to $(0.84+0.86+0.87+0.85+0.87+0.86)-6\times 0.875=-0.1$. This situation is however reversed for $\gamma=0.85$, because q becomes 0.05 for R' and is now larger than that for R ($q=0.89-0.85=0.04$). Size bias occurs at $\gamma=0.85$ in this example since the highest edge weight w_{12} is omitted from R' for v_1. This suggests, in the presence of size bias, community detection may allocate less similar sequences to the same raw cluster and highly similar sequences to distinct raw clusters. Another problem is that parameter γ is still a uniform cluster cut-off threshold analogous to sequence identity threshold T. Nevertheless, community detection is still an effective means to identify potential vertices for individual graph clusters provided these shortcomings are addressed. Indeed, ALFATClust minimizes the impact of size bias by deriving the graph clusters iteratively and replacing them with individual vertices through graph contraction. Also, the subsequent binning process facilitates an adaptive cut-off threshold for individual clusters.

Binning process

The binning cut-off criteria requires calculating the arithmetic mean of the intra-cluster edge weight for every validated graph cluster obtained in the previous iteration. Since a vertex v_i in the contracted graph G represents a validated graph cluster, the average intra-cluster edge weight is regarded as the vertex weight of v_i and is denoted as w_{ii} of W. All diagonal elements w_{ii} of W therefore represent the vertex weights for G and other elements w_{ij} $(i\neq j)$ denote the edge weights. This does not affect the calculation of Eq. (1) for the community detection process because G has no self-loop e_{ii} and thus w_{ii} is not associated with any edge. Initially each vertex v_i in G is simply a singleton graph cluster, and $w_{ii}=1$ for any singleton cluster represented by v_i. For the first iteration ($\gamma=\gamma_{high}$), all raw clusters become graph clusters directly without binning. This is because γ_{high} is supposed to be close to the highest possible edge weight (i.e. 1) so that the raw clusters identified are robust against size bias, and the intra-cluster edge weights are sufficiently large for raw clusters to qualify as graph clusters. The binning process, which is performed individually for each non-singleton raw cluster, begins by sorting its intra-cluster edges in a descending order of edge weight. The two vertices v_i and v_j associated with edge e_{ij} are added to a vertex list J following the sorted edge order, however only the vertex not present in J can be added. When both v_i and v_j do not appear in J, the one representing the larger underlying graph cluster is added first. The first bin is created using the first vertex in *J*. Starting from the second vertex, the selected vertex v_t is eligible to be assigned to an existing bin *B* satisfying Eq. (2):

$$I_{bin}(B,W) - I(B,W,v_t) < I(B,W,v_t) - \gamma_{low} \tag{2}$$

where

$$I_{bin}(B,W) = \begin{cases} w_{ii} & \text{if } |B| = 1 \text{ and } v_i \in B \\ \frac{\sum_{v_i \in B} (\rho(v_i)(\rho(v_i)-1)/2) w_{ii} + \sum_{v_i, v_j \in B, i<j} \rho(v_i)\rho(v_j) w_{ij}}{\sum_{v_i \in B} \rho(v_i)(\rho(v_i)-1)/2 + \sum_{v_i, v_j \in B, i<j} \rho(v_i)\rho(v_j)} & \text{otherwise} \end{cases}$$

$$I(B,W,v_t) = \frac{\sum_{v_i \in B} \rho(v_i) w_{it}}{\sum_{v_i \in B} \rho(v_i)}$$

$\rho(v_i)$ returns the underlying graph cluster size of v_i as the total number of primitive vertices (i.e. vertices in initial *G*) belonging to that cluster. In Fig. 1A, $I_{bin}(B, W)$ denotes the current cluster compactness of *B* in terms of its average intra-bin edge weight. $I(B, W, v_t)$ refers to the average edge weight between v_t and vertices in *B*. By assuming the homologous sequences have significantly higher edge weights (lower Mash distances) with each other than with other non-homologous sequences, v_t should not be assigned to *B* when $I(B, W, v_t)$ is much smaller than $I_{bin}(B, W)$, or it might introduce outliers into *B* otherwise. $I_{bin}(B, W) - I(B, W, v_t)$ in Eq. (2) is therefore an estimate for the cluster separation between *B* and v_t, and $I(B, W, v_t) - \gamma_{low}$ is the cut-off threshold determined by both *B* and v_t as well as the pre-defined γ_{low}. Figure 1A illustrates both cluster separation

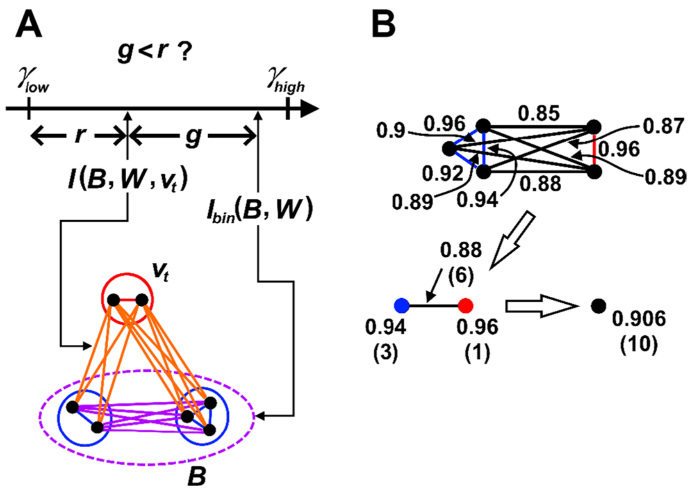

Fig. 1 Illustration of the binning process and graph contraction in ALFATClust. **A** Suppose the black dots are the primitive graph vertices. The binning process determines whether a vertex v_t (red circle) representing a graph cluster of two primitive vertices can be assigned to the bin *B* (purple dashed ellipse) consisting of two vertices (blue circles) containing clusters of sizes 2 and 3. $I_{bin}(B, W)$ is the average edge weight for all edges inter-connecting primitive vertices inside *B* (blue and purple lines) and $I(B, W, v_t)$ refers to the average edge weight for all edges (orange lines) connecting v_t and *B*. The cluster separation $g = I_{bin}(B, W) - I(B, W, v_t)$ and the cluster cut-off threshold $r = I(B, W, v_t) - \gamma_{low}$. According to Eq. (2), v_t can be assigned to *B* when *g* is less than *r*. **B** The graph contraction shrinks two clusters (primitive vertices connected by either blue or red edges) into two vertices (blue and red dots respectively). The intra-cluster (blue and red) and inter-cluster (black) edge weights are averaged to become vertex (blue and red) weights and collapsed edge (black) weights respectively. The value in the parentheses denotes the number of underlying actual edges. These two vertices can be further collapsed into a single vertex (black)

and adaptive cut-off threshold in the graph. v_t is assigned to the bin giving the highest positive score calculated with a scoring function $Q(B, W, v_t)$:

$$Q(B, W, v_t) = 2I(B, W, v_t) - I_{bin}(B, W) - \gamma_{low} \quad (3)$$

The inequality in Eq. (2) is satisfied when $Q(B, W, v_t) > 0$. v_t is assigned to a new bin when none of the scores for the existing bins are positive. As a result, every bin derived from a raw cluster becomes a validated graph cluster for the current iteration.

Graph contraction

After binning all raw clusters, G is contracted such that each graph cluster C_i in G is now replaced by a new vertex v'_i, and its vertex weight w'_{ii} is equal to the average intra-cluster edge weight of C_i. Edges connecting between clusters C_i and C_j are replaced by a single edge connecting v'_i and v'_j in the contracted graph, with its edge weight averaged from the replaced edges. Figure 1B illustrates an example of the graph contraction process. However, graph clusters obtained from the same raw cluster are prohibited from appearing together in any raw cluster in subsequent iterations. This is achieved by assigning a large negative edge weight ($-|V'|^2$ where V' is the vertices of the contracted G) to the edges interconnecting them. The next iteration, if any, begins with the contracted graph. The graph contraction therefore preserves the validated graph clusters in subsequent iterations against size bias, provided that Δ is sufficiently small.

Selection of the core parameters in ALFATClust

The value of γ_{high} is supposed to approach the maximum possible edge weight value (i.e. 1) so that the raw clusters detected in the first iteration are robust against size bias. Most of the actual cut-offs for individual clusters are expected to appear above γ_{low}, which still needs to be set as high as possible to avoid capturing subsequent trivial (but larger) drops. For example, a decrease of edge weight from 0.85 to 0.72 is more significant than the next even larger drop from 0.74 to 0.59. The smaller the value of Δ the lesser is the size bias for CPM in an iteration. The minimum value for γ_{high} is at least 0.95. The value for Δ should not exceed 0.025. Both γ_{high} and Δ are usually relatively invariant and the value of γ_{low} can set between 0.7 and 0.8.

Scalability improvement

Although ALFATClust computes a full Mash distance matrix for its graph clustering, the matrix can be significantly reduced using a divide-and-conquer approach. A pre-clustering step is performed at the beginning to divide the sequences into multiple sequence partitions. This is achieved by running the highly scalable MMSeqs2 with sequence identity threshold T equal to γ_{low}, and each of its output sequence clusters becomes an individual sequence partition. ALFATClust can then run with each partition separately without compromising the overall clustering accuracy. This is because the pairwise sequence similarity between any two sequences in different partitions is below γ_{low}, hence filtering their corresponding vertex pair brings little impact to both CPM scoring community detection and the binning process. ALFATClust runs in pre-clustering mode when either the number of sequences exceeds a pre-defined limit (20,000 sequences by default) or the pre-clustering option is activated by the user.

Cluster evaluation

ALFATClust provides an optional evaluation of cluster quality in terms of sequence identity. For each non-singleton cluster, the sequence giving the largest sum of intra-cluster edge weight and satisfying the sequence length criteria (i.e. between lower quartile − 1.5 × interquartile range and upper quartile + 1.5 × interquartile range) is selected as the representative center sequence. If no such sequence exists or the cluster consists of only two sequences, then the longest sequence will be selected instead. Pairwise sequence identity (number of matched bases divided by alignment length excluding terminal gaps, refer to Additional file 1 for implementation details) is calculated between the center sequence and every other sequence in the same cluster. The evaluation report includes both mean sequence identity and minimum sequence identity for every non-singleton cluster.

Results

Benchmark datasets and setup

Antimicrobial resistance (AMR) gene and protein sequence datasets [32–34] are suitable for evaluating clustering effectiveness due to the presence of many distinct classes of homologous resistance gene sequences, and a wide range of sequence similarities among these sequences. AMR gene sequences are retrieved from various public AMR databases including CARD (version 3.0.7) [32], ResFinder (downloaded on 3th Feb, 2020) [33], and ARG-ANNOT (version 4) [34]. All AMR sequences collected are validated, integrated, annotated into an AMR gene sequence dataset using ARGDIT [35], which further translates this dataset into an AMR protein sequence dataset. Non-AMR plasmid nucleotide sequences are extracted with their original annotation from PLSDB [36] (version 2019_10_07) to create a plasmid nucleotides dataset. Due to its highly variable sequence lengths, it can be used to investigate whether the clustering performance is affected when the sequence length differs substantially. Finally, scalability in terms of the number of sequences and sequence length is examined using viral nucleotide and amino acid sequence datasets (each consisting of ~ 470,000 sequences) retrieved from viruSITE (release 2021.1) [37]. Sequences consisting of ambiguous nucleotide or amino acid, or having their sequence length shorter than the smallest Mash k-mer size benchmarked (13 for gene and 9 for protein) are discarded. In particular, viral nucleotide sequences are split into two separate datasets according to the 10,000 nucleotides criteria. The viral sequence datasets are only used for scalability assessments. Table 1 summarizes the properties of all benchmark datasets used, with further details provided in the "Results" section.

The performance of ALFATClust is compared with CD-HIT (version 4.8.1), UCLUST (version 11.0.667), VSEARCH (version 2.14.1), DNACLUST (version 3.7), MeShClust (also MeShClust^2 [38] for the viral nucleotide dataset), and MMseqs2 (version 12.113e3). The identity thresholds used for these tools range from 0.7 to 0.9 with a step size of 0.05. Any cluster optimization option(s) in the tools are also turned on whenever available except for the viral sequence datasets. The suggested value of γ_{low} is in the range of 0.7

Table 1 Details of the benchmark datasets used for evaluation

Dataset	No. of sequences	Sequence length		
		Mean (standard deviation)	Min	Max
AMR genes	4027	939.93 (± 381.98)	162	4359
AMR proteins	3891	312.53 (± 127.90)	53	1452
Plasmid nucleotides	5005	1010.38 (± 1 008.45)	77	9511
Viral nucleotides	478,652	717.09 (± 837.21)	13	9993
Long viral nucleotides	676	14,803.87 (± 12 048.56)	10,002	262,388
Viral amino acids	469,835	242.64 (± 313.29)	9	13,556

to 0.8 for most instances; in this benchmark γ_{low} is set to three different values: 0.7, 0.75, and 0.8. Default values are applied for other parameters: $\gamma_{high} = 0.95$, and $\Delta = 0.025$. Also, the default Mash k-mer size (17 for nucleotide sequences and 9 for protein sequences) and sketch size (2000) are used except for both plasmid and viral nucleotide datasets where the k-mer size is set to 13. Execution commands and options for all the tools are provided in Additional file 1. Silhouette scores [12] for the sequence clusters are computed using scikit-learn [39], while the results are plotted using Matplotlib [40]. All the benchmarks are performed on a workstation with a quad-core Intel Xeon W-2102 CPU and 64 GB RAM. Cluster evaluation reports of ALFATClust for the AMR and plasmid sequence datasets are available in Additional files 2, 3, 4, 5, 6, 7, 9, 10, 11. In addition, although the Leiden algorithm involves random vertex and community selection, identical set of clusters is obtained from 50 individual runs for each of the AMR and plasmid datasets using any of the three values of γ_{low} specified above.

Sensitivity of clustering parameter γ_{low} and cluster robustness

Figure 2 shows the rate of decrease in number of clusters generated is slower with decreasing γ_{low} in ALFATClust than decreasing T in other clustering approaches, especially for the AMR sequence datasets. One possible explanation is the clustering results being less sensitive to γ_{low}, at least for its suggested value range (0.7–0.8), than T in other clustering tools. A further verification is therefore performed by comparing the clusters derived with these values of γ_{low} and T (0.9, 0.85, and 0.8) (Additional file 1: Figures S6–S12). ALFATClust exhibits the highest proportion of identical clusters shared across different γ_{low}. For example, in the AMR gene dataset, the number of identical clusters common across the three values of γ_{low} is 995 (Additional file 1: Figure S6), which constitute 83.4% of all distinct clusters seen. This proportion is remarkably higher than all other tools such as UCLUST (68.5%, Additional file 1: Figure S8) and MMseqs2 (65.9%, Additional file 1: Figure S9). This better cluster pattern convergence indicates a lower impact of selecting a non-optimal γ_{low} from the suggested range. Another observation is the uneven distribution of the cluster sizes. When running ALFATClust with $\gamma_{low} = 0.75$, the average number of sequences per cluster and the standard deviation for the AMR gene, AMR protein, and plasmid nucleotides datasets are 3.72 ± 13.77, 3.74 ± 13.98, and 7.27 ± 15.66 respectively. Their cluster sizes are therefore highly varied. Moreover, while there are a few clusters consisting of over 100 sequences in each of these datasets, singleton clusters occupy ~ 70% of all derived clusters for the AMR sequence datasets

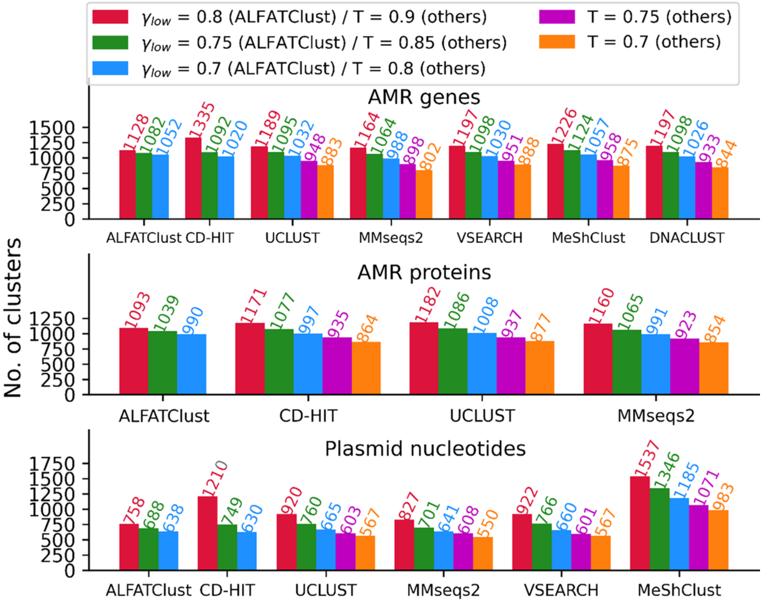

Fig. 2 Number of sequence clusters derived by the clustering tools. γ_{low} for ALFATClust varies from 0.8 down to 0.7 and *T* for others varies from 0.9 down to 0.7, both with a step size of 0.05. The values of γ_{low} or *T* are distinguished by different colours. Only ALFATClust, CD-HIT, UCLUST, and MMseqs2 can cluster protein sequences. The lowest value of *T* allowed for nucleotide sequences by CD-HIT is 0.8. DNACLUST cannot cluster some plasmid nucleotide sequences that are too short

and ~ 39% for the plasmid dataset. These datasets are thus real examples illustrating the difficulty in searching for suitable clustering parameter value (such as *K* in K-means clustering) evaluated with internal validation index.

The distribution of the minimum sequence identities presented in the ALFATClust cluster evaluation reports (available in Additional files 2, 3, 4, 5, 6, 7, 9, 10, 11) is shown in Fig. 3. For $\gamma_{low} \geq 0.75$, at least 74% of the non-singleton clusters have minimum sequence identity ≥ 0.9 in all three benchmark datasets; and only a few clusters have minimum sequence identity below 0.8. The particularly low sequence identities (< 0.7) for the plasmid nucleotides dataset are partially due to multiple large non-terminal gaps present in the pairwise sequence alignment, such as the one between the cluster center sequence "CP016074.1_rep7a_15_repC(pS0385p1)" and sequence "NC_017335.1_rep7a_18_rep(pS0385p2)" in the same cluster. Another reason is the significantly underestimated Mash distance due to partially overlapping sequence segments. Plasmid nucleotide sequence pair "LT906556.1_IncFII(pCoo)_1_pCoo" and "KX276657.1_IncFIC(FII)_1" is an example demonstrating such partial overlap, with the Mash distance equal to 0.069 (hence the corresponding sequence similarity is $1 - 0.069 = 0.931$), while the true sequence identity calculated using alignment is only 0.555. The alignments giving large non-terminal gaps or partial overlap for the above examples are illustrated in Additional file 1. By examining the cluster evaluation reports generated, it is found that for the AMR gene dataset with $\gamma_{low} = 0.7$, the sequence pair having the lowest minimum sequence identity belongs to the same AMR gene family (AMR genes LRA-3 and LRA-9, both under subclass B3 LRA beta-lactamase according to CARD). For the AMR protein dataset with $\gamma_{low} = 0.7$, the sequence pair having the lowest identity

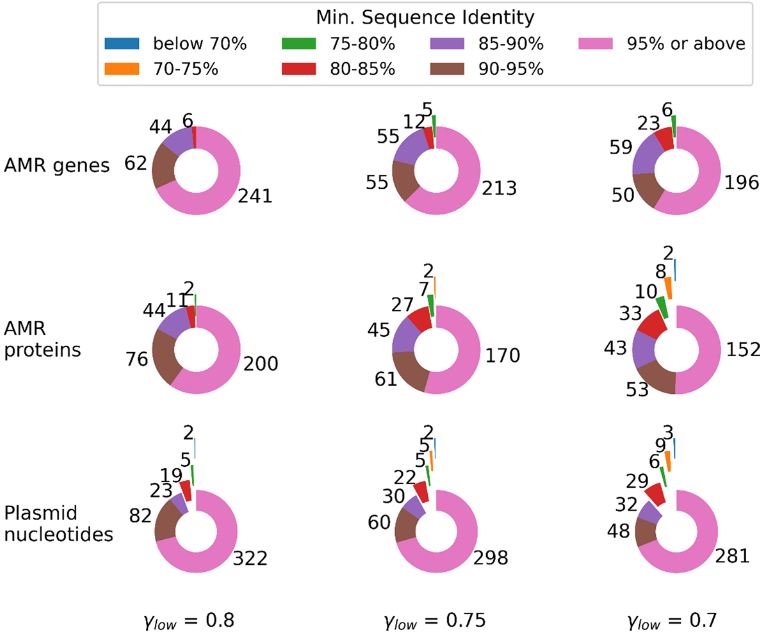

Fig. 3 Distribution of non-singleton sequence clusters based on minimum sequence identity. The minimum sequence identity for a sequence cluster is the lowest sequence identity observed between its representative center sequence and another sequence in the same cluster. The value next to the slice indicates the number of clusters belonging to a particular sequence identity category

score belongs to dihydrofolate reductase (conferring resistance to trimethoprim), although the sequence identity for this pair is quite low (0.675).

Overall sequence cluster quality benchmark

The overall sequence cluster quality of ALFATClust is compared with other approaches using both external and internal validation indices. Normalized mutual information (NMI) [41] and purity [42] are external validation indices used for this comparison with the AMR sequence datasets. The AMR sequences are manually classified into gene classes created using the gene names (and their synonyms provided by CARD [32]) identified from the AMR gene sequence annotations. Sets of AMR sequence clusters obtained by various approaches at different thresholds are evaluated individually against a set of 827 AMR gene classes consisting of 3 720 AMR sequences (accounting for ~ 92% of all AMR gene sequences, refer to Additional file 8 for the AMR gene classes). For a set of sequence clusters L consisting of N sequences together, NMI measures the correlation between L and the pre-defined gene classes Ω using Eq. (4) below:

$$NMI(L, \Omega) = \frac{2 \times I(L; \Omega)}{H(L) + H(\Omega)} \tag{4}$$

where

$$I(L; \Omega) = \sum_{C \in L} \sum_{\omega \in \Omega} \frac{|C \cap \omega|}{N} \log \left(\frac{N|C \cap \omega|}{|C||\omega|} \right)$$

and

$$H(\Theta) = -\sum_{\theta \in \Theta} \frac{|\theta|}{N} \log\left(\frac{|\theta|}{N}\right)$$

The higher the values for NMI (maximum NMI is 1) the better correlation is between the sequence clusters and the gene classes. Figure 4 illustrates how NMI varies with γ_{low} for ALFATClust as well as T for other approaches. The NMI values obtained by ALFATClust are not less than any other tool (Additional file 1: Tables S2 and S3) in the AMR gene dataset, and its NMI at γ_{low}=0.7 is the overall highest (0.933) in the AMR protein dataset. Moreover, the NMI value variation for γ_{low} between 0.7 and 0.8 is relatively small compared to others in both datasets. This is mainly due to the lower sensitivity of the clustering outcomes with γ_{low} than T. Purity is another external index to assess the homogeneity of gene class in the sequence clusters. Higher purity (maximum purity is 1) means the clusters are generally more dominated by sequences in the same gene class. Equation (5) below calculates purity:

$$Purity = \frac{1}{N} \sum_{C \in L} \max_{\omega \in \Omega} |C \cap \omega| \quad (5)$$

Figure 5 shows a strictly decreasing trend of purity with decreasing γ_{low} and T for all the clustering approaches benchmarked. Nevertheless, the rate of decrease for ALFATClust is slower than others, and its purity is often close to or even higher than

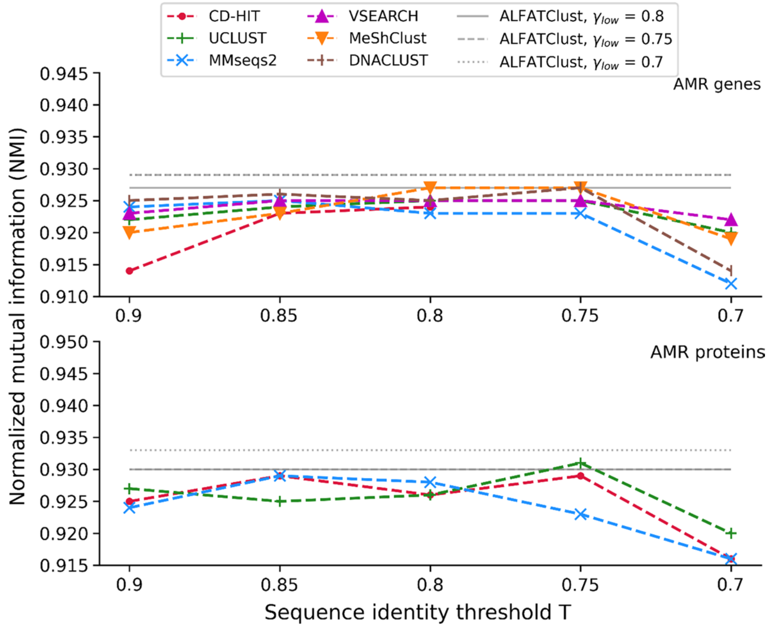

Fig. 4 Comparison of NMI between ALFATClust and other clustering tools. γ_{low} for ALFATClust varies from 0.8 down to 0.7 (solid to dotted lines) and T for other clustering tools (x-axis) varies from 0.9 down to 0.7, both with a step size of 0.05. Only ALFATClust, CD-HIT, UCLUST, and MMseqs2 can cluster protein sequences. The lowest value of T allowed for nucleotide sequences by CD-HIT is 0.8

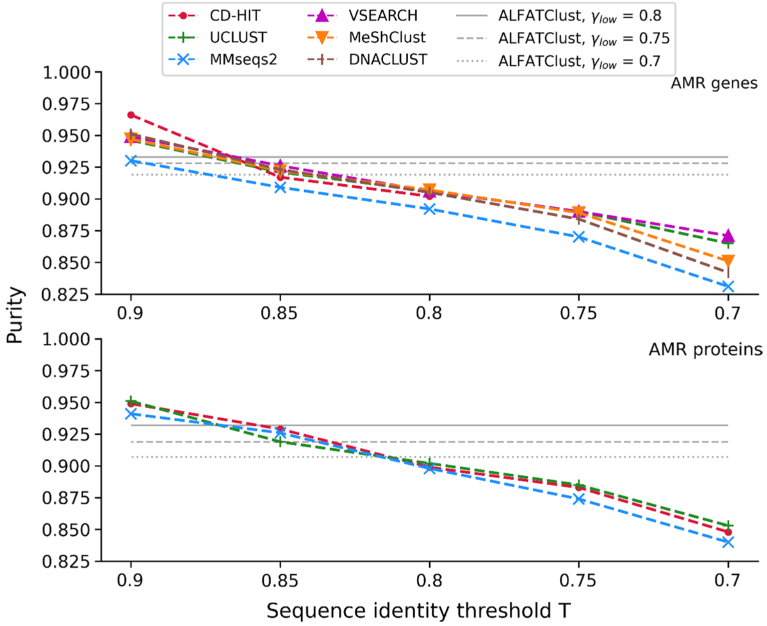

Fig. 5 Comparison of purity between ALFATClust and other clustering tools. γ_{low} for ALFATClust varies from 0.8 down to 0.7 (solid to dotted lines) and *T* for other clustering tools (x-axis) varies from 0.9 down to 0.7, both with a step size of 0.05. Only ALFATClust, CD-HIT, UCLUST, and MMseqs2 can cluster protein sequences. The lowest value of *T* allowed for nucleotide sequences by CD-HIT is 0.8

other tools at $T = 0.85$ (exact values for ALFATClust and other methods are shown in Additional file 1: Tables S4 and S5 respectively). The implication is that the cluster expansion in ALFATClust is less aggressive than the greedy algorithms towards lower thresholds. This is the benefit of determining an individual cut-off for each cluster by considering the separation with its neighbour clusters, rather than relying on a single rigid threshold to allocate sequences.

Internal validation indices such as silhouette coefficient c [12] evaluates the overall cluster separation using the true sequence distance matrix (rather than the Mash distance matrix). c is defined as:

$$c = \frac{1}{|S|} \sum_{s \in S} \frac{d_{neighbor}(s) - d_{intra}(s)}{\max\left(d_{neighbor}(s), d_{intra}(s)\right)} \tag{6}$$

where $d_{neighbor}(s)$ measures the separation of sequence s, i.e. mean distance between s and a sequence in its nearest (called "neighboring") cluster, and $d_{intra}(s)$ measures the cohesion of s, i.e. mean distance between s and other sequences in the same cluster. c ranges from -1 to 1, and a higher value of c indicates a better overall cluster separation. Internal validation indices can be used for cluster quality evaluation when the pre-determined sequence classes are unknown (e.g. the plasmid nucleotides dataset). To compute the true sequence distance matrix for the plasmid nucleotides dataset, exact pairwise sequence identity (same calculation formula as the one used for cluster evaluation in ALFATClust) is calculated for every pair of sequences, and the pairwise distance is given by $1 - \text{sequence identity}$. Figure 6 shows that while the maximum silhouette coefficient

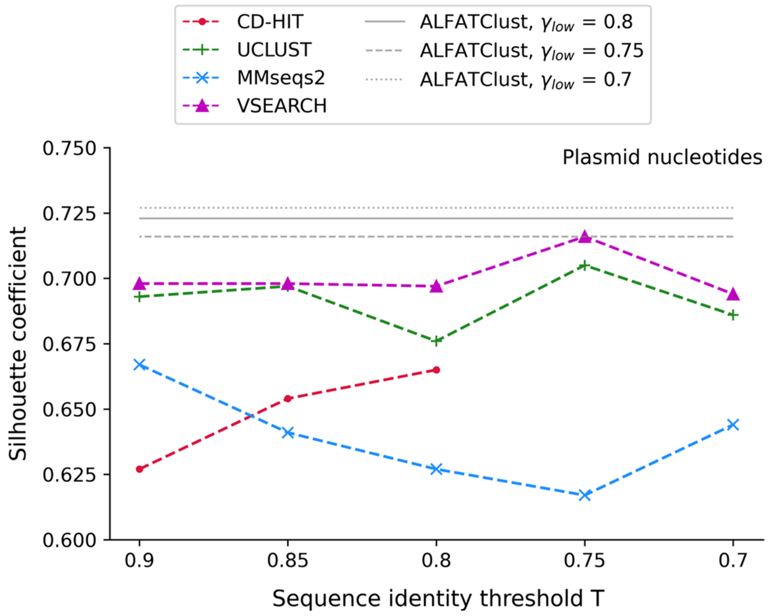

Fig. 6 Comparison of silhouette coefficient between ALFATClust and other clustering tools for the plasmid nucleotides dataset. γ_{low} for ALFATClust varies from 0.8 down to 0.7 (solid to dotted lines) and *T* for other clustering tools (x-axis) varies from 0.9 down to 0.7, both with a step size of 0.05. The lowest value of *T* allowed for nucleotide sequences by CD-HIT is 0.8. DNACLUST cannot cluster some plasmid nucleotide sequences that are too short, and the silhouette coefficients for MeShClust are below zero for the plasmid nucleotides dataset with all values of *T*

values are attained at distinct values of *T* for different tools, none of them is better than those for ALFATClust at all three values of γ_{low}. By comparing the silhouette coefficients between Additional file 1: Tables S6 and S7, it can be seen that the lowest silhouette coefficient achieved (0.716) by ALFATClust is equal to the highest score obtained with other approaches for the plasmid dataset.

Scalability performance

The scalability benchmark is performed with respect to the number of sequences using the viral nucleotide and amino acid sequence datasets, and sequence length using the long viral nucleotide dataset. ALFATClust clusters these datasets with $\gamma_{low}=0.75$ (without the optional cluster evaluation), while others perform clustering with $T=0.85$. The processing times shown in Table 2 vary substantially among different clustering approaches, ranging from a minute or less to several hours. Both MMSeqs2 and MeShClust runs faster than ALFATClust for the viral nucleotide dataset, but the number of clusters derived by MeShClust is unusually low. UCLUST requires less than a minute to cluster ~ 470,000 viral amino acid sequences, but it cannot process long viral nucleotide sequences due to its software limitation. MMseqs2 as well as the alignment-free ALFATClust and MeShClust^2 run much faster than other alignment-based tools for long viral nucleotide sequences. In summary, ALFATClust is scalable for a large number of sequences due to the efficient pre-clustering based on MMseqs2, and sequence length through alignment-free sequence distance calculation.

Table 2 Scalability benchmark through the viral nucleotide and amino acid sequence datasets

	Datasets					
	Viral nucleotides ($\leq$ 10,000 nts)		**Long viral nucleotides (> 10,000 nts)**		**Viral amino acids[@]**	
	No. of clusters	**Time (hh:mm:ss)**	**No. of clusters**	**Time (hh:mm:ss)**	**No. of clusters**	**Time (hh:mm:ss)**
ALFATClust[$]	237,276	00:35:41	499	00:00:39	234,451	00:27:23
CD-HIT	N.A.[#]		506	00:16:10	109,515	04:29:06
UCLUST	245,658	07:47:25	N.A.*		243,982	00:00:56
MMseqs2	221,997	00:02:18	428	00:00:15	235,921	00:06:43
VSEARCH	239,728	05:23:11	507	01:48:36		
MeShClust	8713	00:14:53	462	01:14:52		
MeShClust^2	194,267	03:43:00	571	00:00:10		
DNACLUST	N.A.^		N.A.^			

[$] ALFATClust runs at $\gamma_{low} = 0.75$, and all other tools run at $T = 0.85$

[#] Terminated after running for 8 h

*Memory limit exceeded for the community (32-bit) version of UCLUST

^Segmentation fault occurs

[@] Only ALFATClust, CD-HIT, UCLUST, and MMseqs2 can process protein sequences

Discussion

ALFATClust is conceptually similar to hierarchical agglomerative clustering since its algorithm begins with each sequence (vertex) as a singleton graph cluster, and the graph clusters are gradually merged through iterations with decreasing resolution parameter γ. The graph contraction at the end of each iteration preserves the integrity of the current graph clusters against the size bias of CPM in subsequent iterations. Moreover, for each raw cluster, vertex sorting prioritizes its vertex pairs in descending order of edge weight (i.e. ascending order of sequence distance) for subsequent binning. Both intra-cluster and inter-cluster edge weight calculations are based on unweighted average linkage as shown in Fig. 1B. Using the CPM formulated in Eq. (1), community detection (Leiden algorithm) acts as a selection process at a particular value of γ to identify potential vertices for individual graph clusters. Equation (3) is the scoring function proposed to bin these vertices within each raw cluster to one or more graph clusters. For any existing bin B, it considers both the average intra-bin edge weight $I_{bin}(B, W)$, which is equal to the mean sequence similarity inferring sequence homology, and the proximity between B and vertex v_t, i.e. $I(B, W, v_t)$, with respect to γ_{low}. Note that this scoring function only depends on the constant γ_{low} but not the variable γ in its calculation, hence the binning process is consistent throughout iterations. Although it references a single value of γ_{low}, every bin has its own actual cut-off above γ_{low}. In other words, γ_{low} is a soft cut-off as opposed to a global hard cut-off like sequence identity threshold T or number of clusters K [13] to define all output clusters. Individual adaptive cluster cut-offs offer greater flexibility to fit different clusters such that the cut-off value can be lowered for only certain clusters when necessary, i.e. to expand a cluster by including those slightly less similar sequences, if any. The benchmark analysis suggests that the clustering outcomes are less sensitive to different values of γ_{low} (at least for those within the suggested value range) compared to T, therefore reducing the impact of selecting a non-optimal soft cut-off. It is also easier to simultaneously maintain relatively high cluster quality and cluster separation for a wider range of γ_{low}. This is much harder for other algorithms to balance between these two criteria through a uniform cluster cut-off threshold, of which the optimal value is often unknown and difficult to determine. Moreover, ALFATClust is scalable towards sequence length due to the use of alignment-free sequence distance calculation such as Mash.

From the sequence similarity point of view, both the iterative cluster computation from γ_{high} down to γ_{low} (provided Δ is sufficiently small) accompanied by graph contraction, and the vertex sorting prior to the binning process generally prioritize more similar sequence pairs for clustering. This is particularly important because the linear correlation between the average nucleotide identity (ANI) and the Mash distance d decreases with increasing d [25], and thus this prioritization allows the sequence pairs to be processed from the most reliable (and small) sequence distances. In addition, since γ_{low} determines the value of γ for the final iteration, most of the pairwise sequence similarity values below γ_{low} are actually filtered, and so the clustering outcomes are not affected even they deviate substantially from the true sequence identities. Mash distance inaccuracies are mitigated by averaging the edge weights when collapsing the relevant vertices. The proposed ALFATClust algorithm allows pre-clustering to enhance scalability towards number of sequences without disrupting the overall clustering outcomes, because the sequences from distinct partitions are supposed to be dissimilar with each other. Cluster evaluation is also available for users to inspect the quality of the non-singleton sequence clusters, and determine whether the specified value of γ_{low} is appropriate for the given dataset. In particular, the recommended value for γ_{low} is 0.7 or above, because the linear correlation between ANI and d might become distorted when its value is too low. It is sufficient to set γ_{high} to 0.95 for most cases, and the step size Δ is not larger than 0.025.

Although both ALFATClust and CARNAC-LR [43] are clustering tools based on community detection, they do have fundamental differences. Firstly, CARNAC-LR relies on read alignment (mapping) to determine whether two long reads overlap significantly based on a similarity threshold set in the alignment tool, and an unweighted edge between two vertices (reads) denotes such overlap; ALFATClust creates a complete graph using the alignment-free Mash distances, which are converted to edge weights representing (average) pairwise sequence similarities. Hence, it does not impose any similarity threshold for graph construction. Secondly, CARNAC-LR partitions the graph into K cliques or dense subgraphs by minimizing the number of edges between them, and the value of K is determined internally; ALFATClust performs sequence clustering based on a user-specified soft cut-off γ_{low} as explained above. Finally, CARNAC-LR resolves intersecting clusters by identifying spurious read overlaps and cutting the edges accordingly; ALFATClust gradually expands the clusters from the largest (and also the most reliable) edge weights first to minimize the impact of sequence distance approximation errors.

ALFATClust inherits the limitations of Mash distance, particularly distance underestimation for partially overlapping sequences such as the plasmid sequence cases discussed in the “Results” section. Therefore, when the input sequences consist of genome sequence fragments rather than complete gene sequences, users are advised to identify clusters with particularly low sequence identities through the evaluation report to detect potential partial overlaps. Moreover, compared to genome-scale sequences, gene sequences are more sensitive to the specified k-mer size for accurate distance calculation, and this value is shown to be varying between different datasets. It should also be noted that ALFATClust is intended to be used to partition multiple groups of homologous sequences, it is therefore not suitable for tasks such as OTU (operational taxonomic unit) clustering, in which the sequence identity threshold required is strictly 97%.

Conclusions

Our benchmark demonstrates numerous advantages of ALFATClust over typical threshold-based sequence clustering approaches, including better clustering results for a non-optimal soft cut-off threshold, generally large cluster separation, and scalability with respect to number of sequences and sequence length. It also facilitates cluster quality inspection by providing cluster evaluation. It is suitable for clustering multiple groups of homologous sequences in which the sequence similarity cut-off threshold is often unknown and hard to determine.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-022-04643-9.

Additional file 1. Supplementary method and benchmark details, figures, and tables.

Additional file 2. ALFATClust cluster evaluation report for the AMR gene sequence dataset ($\gamma_{\text{low}} = 0.7$).

Additional file 3. ALFATClust cluster evaluation report for the AMR gene sequence dataset ($\gamma_{\text{low}} = 0.75$).

Additional file 4. ALFATClust cluster evaluation report for the AMR gene sequence dataset ($\gamma_{\text{low}} = 0.8$).

Additional file 5. ALFATClust cluster evaluation report for the AMR protein sequence dataset ($\gamma_{\text{low}} = 0.7$).

Additional file 6. ALFATClust cluster evaluation report for the AMR protein sequence dataset ($\gamma_{\text{low}} = 0.75$).

Additional file 7. ALFATClust cluster evaluation report for the AMR protein sequence dataset ($\gamma_{\text{low}} = 0.8$).

Additional file 8. Gene classification for the AMR gene sequence dataset. Used as the background truth for the NMI and purity evaluation.

Additional file 9. ALFATClust cluster evaluation report for the plasmid nucleotide sequence dataset ($\gamma_{\text{low}} = 0.7$).

Additional file 10. ALFATClust cluster evaluation report for the plasmid nucleotide sequence dataset ($\gamma_{\text{low}} = 0.75$).

Additional file 11. ALFATClust cluster evaluation report for the plasmid nucleotide sequence dataset ($\gamma_{\text{low}} = 0.8$).

Acknowledgements
Not applicable.

Authors' contributions
JKHC and RTHO conceived the method and wrote the manuscript. JKHC implemented the software and performed the experiments. Both authors read and approved the final manuscript.

Funding
This research is supported by the Saw Swee Hock School of Public Health, National University of Singapore under its Infectious Disease Programme Research (SSHSPH ID-PRG/PILOTGRANT/2018/04).

Availability of data and materials
ALFATClust is available at https://github.com/phglab/ALFATClust. AMR sequence and plasmid datasets are also available in this package.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 26 March 2021 Accepted: 2 March 2022
Published online: 30 March 2022

References

1. Murtagh F, Contreras P. Algorithms for hierarchical clustering: an overview. WIREs Data Min Knowl Discov. 2012;2(1):86–97.
2. National Center for Biotechnology Information (NCBI): Documentation of the BLASTCLUST-algorithm. ftp://ftp.ncbi.nih.gov/blast/documents/blastclust.html.
3. Enright AJ, Ouzounis CA. GeneRAGE: a robust algorithm for sequence clustering and domain detection. Bioinformatics. 2000;16(5):451–7.
4. Loewenstein Y, Portugaly E, Fromer M, Linial M. Efficient algorithms for accurate hierarchical clustering of huge datasets: tackling the entire protein space. Bioinformatics. 2008;24(13):i41–9.
5. Uchiyama I. Hierarchical clustering algorithm for comprehensive orthologous-domain classification in multiple genomes. Nucleic Acids Res. 2006;34(2):647–58.
6. Lloyd S. Least squares quantization in PCM. IEEE Trans Inf Theory. 1982;28(2):129–37.
7. MacQueen J. Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol 1: statistics: 1967 1967; Berkeley, Calif.: University of California Press. pp. 281–297.
8. Ashlock D, Warner E. Classifying synthetic and biological DNA sequences with side effect machines. In: 2008 IEEE symposium on computational intelligence in bioinformatics and computational biology: 15-17 Sept. 2008 2008. pp. 22–29.
9. Kelarev A, Kang B, Steane D. Clustering algorithms for ITS sequence data with alignment metrics. Lect Notes Comput Sci. 2006;4304:1027–31.
10. Boratyn GM, Camacho C, Cooper PS, Coulouris G, Fong A, Ma N, Madden TL, Matten WT, McGinnis SD, Merezhuk Y, et al. BLAST: a more efficient report with usability improvements. Nucleic Acids Res. 2013;41(W1):W29–33.
11. Liu Y, Li Z, Xiong H, Gao X, Wu J, Wu S. Understanding and enhancement of internal clustering validation measures. IEEE Trans Cybern. 2013;43(3):982–94.
12. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53–65.
13. Wei D, Jiang Q, Wei Y, Wang S. A novel hierarchical clustering algorithm for gene sequences. BMC Bioinform. 2012;13(1):174.
14. Fu L, Niu B, Zhu Z, Wu S, Li W. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics. 2012;28(23):3150–2.
15. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics. 2006;22(13):1658–9.
16. Edgar RC. Search and clustering orders of magnitude faster than BLAST. Bioinformatics. 2010;26(19):2460–1.
17. Ghodsi M, Liu B, Pop M. DNACLUST: accurate and efficient clustering of phylogenetic marker genes. BMC Bioinform. 2011;12(1):271.
18. Rognes T, Flouri T, Nichols B, Quince C, Mahé F. VSEARCH: a versatile open source tool for metagenomics. PeerJ. 2016;4:e2584.
19. Li W, Jaroszewski L, Godzik A. Clustering of highly homologous sequences to reduce the size of large protein databases. Bioinformatics. 2001;17(3):282–3.
20. Li W, Jaroszewski L, Godzik A. Sequence clustering strategies improve remote homology recognitions while reducing search times. Protein Eng Des Sel. 2002;15(8):643–9.
21. James BT, Luczak BB, Girgis HZ. MeShClust: an intelligent tool for clustering DNA sequences. Nucleic Acids Res. 2018;46(14):e83–e83.
22. Cheng Y. Mean shift, mode seeking, and clustering. IEEE Trans Pattern Anal Mach Intell. 1995;17(8):790–9.
23. Steinegger M, Söding J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol. 2017;35:1026.
24. Steinegger M, Söding J. Clustering huge protein sequence sets in linear time. Nat Commun. 2018;9(1):2542.
25. Ondov BD, Treangen TJ, Melsted P, Mallonee AB, Bergman NH, Koren S, Phillippy AM. Mash: fast genome and metagenome distance estimation using MinHash. Genome Biol. 2016;17(1):132.
26. Baker DN, Langmead B. Dashing: fast and accurate genomic distances with HyperLogLog. Genome Biol. 2019;20(1):265.
27. Traag VA, Waltman L, van Eck NJ. From Louvain to Leiden: guaranteeing well-connected communities. Sci Rep. 2019;9(1):5233.
28. Girvan M, Newman MEJ. Community structure in social and biological networks. Proc Natl Acad Sci. 2002;99(12):7821–6.
29. Traag VA, Van Dooren P, Nesterov Y. Narrow scope for resolution-limit-free community detection. Phys Rev E. 2011;84(1):016114.
30. Blondel VD, Guillaume J-L, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. J Stat Mech Theory Exp. 2008;2008(10):P10008.

31. Jones I, Wang R, Han J, Liu H. Community cores: removing size bias from community detection. In: Proceedings of the international AAAI conference on web and social media 2016, 10(1).
32. Alcock BP, Raphenya AR, Lau TTY, Tsang KK, Bouchard M, Edalatmand A, Huynh W, Nguyen A-LV, Cheng AA, Liu S, et al. CARD 2020: antibiotic resistome surveillance with the comprehensive antibiotic resistance database. Nucleic Acids Res. 2019;48(1):517–25.
33. Zankari E, Hasman H, Cosentino S, Vestergaard M, Rasmussen S, Lund O, Aarestrup FM, Larsen MV. Identification of acquired antimicrobial resistance genes. J Antimicrob Chemother. 2012;67(11):2640–4.
34. Gupta SK, Padmanabhan BR, Diene SM, Lopez-Rojas R, Kempf M, Landraud L, Rolain J-M. ARG-ANNOT, a new bioinformatic tool to discover antibiotic resistance genes in bacterial genomes. Antimicrob Agents Chemother. 2014;58(1):212–20.
35. Chiu JKH. Ong RT-H: ARGDIT: a validation and integration toolkit for antimicrobial resistance gene databases. Bioinformatics. 2019;35(14):2466–74.
36. Galata V, Fehlmann T, Backes C, Keller A. PLSDB: a resource of complete bacterial plasmids. Nucleic Acids Res. 2018;47(D1):D195–202.
37. Stano M, Beke G, Klucar L. viruSITE—integrated database for viral genomics. Database 2016; 2016.
38. James BT, Girgis HZ: MeShClust2: Application of alignment-free identity scores in clustering long DNA sequences. bioRxiv 2018:451278.
39. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30.
40. Hunter JD. Matplotlib: a 2D graphics environment. Comput Sci Eng. 2007;9(3):90–5.
41. Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J Mach Learn Res. 2010;11:2837–54.
42. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. Cambridge: Cambridge University Press; 2008.
43. Marchet C, Lecompte L, Silva CD, Cruaud C, Aury J-M, Nicolas J, Peterlongo P. De novo clustering of long reads by gene from transcriptomics data. Nucleic Acids Res. 2018;47(1):e2–e2.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.