Genealogical inference and more flexible sequence clustering using iterative PopPUNK

Bin Zhao[1], John A. Lees[2,3], Hongjin Wu[1], Chao Yang[1*], Daniel Falush[1]*

[1]The Center for Microbes, Development and Health, CAS Key Laboratory of Molecular Virology and Immunology, Institut Pasteur of Shanghai, Chinese Academy of Sciences, Shanghai, China

[2]MRC Centre for Global Infectious Disease Analysis, School of Public Health, Imperial College London, London W2 1PG, UK

[3]European Molecular Biology Laboratory, European Bioinformatics Institute EMBL-EBI, Hinxton CB10 1SD, UK

*Corresponding author daniel.falush@ips.ac.cn, cyang@ips.ac.cn

Abstract

Bacterial genome data is accumulating at an unprecedented speed due the routine use of sequencing in clinical diagnoses, public health surveillance and population genetics studies. Genealogical reconstruction in fundamental to many of these uses, however, inferring genealogy from large-scale genome datasets quickly, accurately, and flexibly is still a challenge. Here, we extend an alignment- and annotation-free method, PopPUNK, to increase its flexibility and interpretability across datasets. Our method, iterative-PopPUNK, rapidly produces multiple consistent cluster assignments across a range of sequence identities. By constructing a partially resolved genealogical tree with respect to these clusters, users can select a resolution most appropriate for their needs.
We demonstrated the accuracy of clusters at all levels of similarity and genealogical inference of iterative-PopPUNK based on simulated data and obtained phylogenetically-concordant results in real datasets from seven bacterial species. Using an example set of Escherichia/Shigella genomes we show that iterative-PopPUNK can achieve cluster resolutions ranging from phylogroup down to sequence typing (ST). The iterative-PopPUNK algorithm is implemented in the 'PopPUNK_iterate' program, available as part of PopPUNK package (https://github.com/bacpop/PopPUNK).

Introduction

A first step in almost any analysis of collections of bacterial genome sequences is to understand how the sequences are related to each other. For example, to make a clinical diagnosis, a key question is which previously characterized organisms are similar to strains present in the patient sample(Gardy and Loman 2018; Gu et al. 2019). To identify transmission chains, it is necessary to establish which of the sequenced strains share very recent common ancestors, indicating a shared source (Croucher and Didelot 2015; Sintchenko and Holmes 2015). To map genotype-phenotype relationships, it is often helpful to delineate lineages. Organisms from the same lineage have similar sequence content and thus are likely to have similar phenotypes(Falush 2016; Power et al. 2017; Brinda et al. 2020). To accurately reconstruct the evolutionary history of a species, it is necessary to obtain a sample of organisms that is broadly representative of the species' diversity as a whole, and is not biased towards particular lineages, for example by containing only multiple closely related sequences from the same environmental source.

A "gold standard" for establishing genetic relationships between strains is through genealogical reconstruction. Any given sample of bacteria are related to each other by a tree, with each node of the tree representing the most recent common ancestor (MRCA) of the strains below it in the genealogy. The genealogical history reflects the clonal relationships between strains; it is unique and unambiguous. Genome sequence data provides partial information on the topology of this tree, its MRCA and where and when that MRCA existed. In practice, the information that genome sequences provide on genealogical relationships is both incomplete and time consuming to extract. Moreover, a full genealogy is rarely necessary for most practical uses of genetic relationships. This has led to the development of methods that attempt to identify clusters of closely related organisms. A simple interpretation of each cluster is that it corresponds to the set of strains descended from a single node of the genealogical tree. Amongst these possible methods, PopPUNK is notable because it works from raw sequence data and does not require a genome alignment(Lees et al. 2019). This means that the clustering can be obtained with significantly less computational resources and is much faster than methods based on sequence alignments. Moreover, PopPUNK is easily extendable, as new batches of genomes can be integrated into an existing database without needing to recalculate all pairwise distances and model fit.

PopPUNK was originally designed to identify a single level of clusters, attempting to make this choice optimal by adapting a novel machine learning algorithm. However, in practice, multiple levels of clusters with different genetic similarities are often required to achieve different purposes. For example, in the disease outbreak scenario, epidemiologists require clusters of very closely related or indistinguishable strains to help identify and track outbreaks. For routine surveillance and subtyping, clusters with higher diversity are required to represent strains from a sequence type or clonal group. Population geneticists usually require clusters of population or sub-population level to conduct species-level studies. Two state-of-the-art pipelines for large-scale genome datasets that allow multi-level clustering are Fastbaps (Tonkin-Hill et al. 2019)and HierCC, as implemented in Enterobase (Zhou et al. 2021). Fastbaps is based on a core genome alignment, which is slow to generate for

datasets of even modest sizes. HierCC requires an appropriate curated database of genes and alleles or a stable strain nomenclature and cannot be the general solution for most bacterial species.

Here we present iterative-PopPUNK – an extension of the PopPUNK software - that allows estimation of a partially resolved genealogical tree and provides clustering consistent with that tree at different levels of sequence similarity. This enhancement provides greater flexibility than the original PopPUNK algorithm, allowing the fineness of the clustering to be adjusted according to the purpose at hand. Moreover, the clusters can be interpreted more readily, by reference to the full (partially resolved) genealogy. The speed of iterative-PopPUNK is similar to PopPUNK, which can complete the analysis of thousands of bacterial genomes in a few minutes, and can scale even to very large-scale genome datasets (e.g. >100,000). Using simulated data, we demonstrate the accuracy of the clusters from iterative-PopPUNK at all output levels of similarity. Furthermore, we show that for real data, the produced clusters are normally highly concordant with those obtained by phylogenetic methods, but without requiring an alignment.

Results

Overview of iterative-PopPUNK

An overview of Iterative-PopPUNK pipeline is shown in Figure 1A. The first two steps proceed as usual using the original PopPUNK algorithm: 1) database construction and distance calculation and 2) model fitting using original PopPUNK algorithm. Iterative-PopPUNK then performs four additional steps to achieve multi-level clustering and generate clusters under the given similarity cutoffs.

Step 3: Multi-level clustering by moving decision boundary iteratively

After an initial model has been fitted, the PopPUNK identifies two fitted components representing “within-strain” comparisons between closely related isolates and “between-strain”. However, unlike the original PopPUNK algorithm, which identifies a single boundary for defining a “best” cluster assignment by optimising a heuristic (the network score), iterative-PopPUNK instead ignores the network score, and places the decision boundary at multiple equally spaced positions perpendicular to the line connecting within- and between-strain components to create multi-level cluster assignments. The first viable boundary point is where at least one non-trivial cluster (containing at least two samples) is formed, corresponding to the two most closely related sequences in the dataset. The number of boundaries, and therefore number of starting sets of clusters can be set by the user, but we found 30 to be a sensible default for most scenarios.

Step 4. Quality control (QC) of clusters obtained at different levels

QC is performed to remove any duplicated or inconsistent clusters. The clusters obtained under the first viable boundary point are directly defined as QC-passed clusters due to their high accuracy based on the evaluation using simulation data. A new cluster assignment C_j (which is a set of samples) at boundary point *j* was compared with QC-passed clusters C_i at the previous boundary *i*, and is considered QC-passed if and only if it meets one of the following conditions, as shown in the conceptual outline in Figure 1B:

1) C_j contains all of isolates from a QC-passed cluster C_i ($C_i \subseteq C_j$, condition 1);

2) C_j is unique and contains none of isolates from any previous C_i ($Cj \cap Ci = \varnothing$, condition 2);
3) C_j is a strict subset of one of the previous QC-passed cluster C_i ($C_j \subseteq C_i$, condition 3);

By iterating through all pairs of clusters between the current step and next step forward (as clusters increase in size; equivalently the boundary moves upwards), a new QC-passed C_j was added into the list C_i at the next round, until the QC of all the clusters is completed.

Step 5. Partially resolved genealogical tree construction and average core distance calculation

A partially resolved genealogical tree is then constructed based on QC-passed clusters by nesting clusters within larger ones (Figure 1D). The algorithm we used to do this is as follows:

1) Place all strains at the root as "cluster zero";
2) Run through the QC-passed clusters, from largest to smallest. For each one:
    a) Look through all clusters which have been added to the tree and find the smallest cluster which is a superset of the cluster being tested;
    b) Nest the new cluster as a child under this parent cluster;
    c) Remove the strains in the child cluster from the parent;
3) For the remaining strains (which may be at parent nodes, or the tips) add in a child node;
4) For each of the nodes in the tree, we calculate the average core distances between the strains under this node and set its branch length relative to root accordingly.

Step 6. Choosing clusters under given similarity cutoffs

Considering the large difference in the genetic diversity of different bacterial species, we designed two types of relative cutoffs to unify the genetic similarity across species: 1) $P_{max\text{-}cluster\text{-}dist}$, the percentage of Maximum value of the Average Core-Distance (MACD) of all the iterative-PopPUNK clusters (extracted from local PopPUNK distance database). 2) $P_{bet\text{-}mean\text{-}dist}$, the percentage of mean pairwise core-distance between all isolates (extracted from initial model fitting results);

When one of these cutoffs is provided, iterative-PopPUNK searches all the nodes except root node in the partially resolved tree and extracts the relative similarity values ($P_{max\text{-}cluster\text{-}dist}$ or $P_{bet\text{-}mean\text{-}dist}$) of all the samples under each node (NP) and its parent node (N+1P). If a node meets the condition NP ≤ cutoff < N+1P, all the child strains under this node are assigned into a cluster (Figure 1C). The remaining unassigned strains were defined as singletons.

Evaluation of iterative-PopPUNK using simulation data

To test the performance of iterative-PopPUNK, we generated seven datasets of simulated genomes where the true genealogy is known. We generated 100 genomes with a length of 1,000 Kb per dataset and their corresponding genealogical trees using FastSimBac (De Maio and Wilson 2017), representing bacteria with different mutation and recombination rates (μ = 0.01-0.06, r=0.003-0.04, Table S1). We created clusters of these simulated genomes using PopPUNK and iterative-PopPUNK using three different types of model fit (DBSCAN and GMM with two and three components), and mapped the clusters obtained back to the simulated genealogical trees.

To assess performance, we compared the consistency of the output clusters to the known clonal relationships from the simulations. If the strains of an inferred cluster also clustered together (were derivatives of a common node) in the true genealogical tree, we defined this as a matched cluster and the corresponding node was colored in red; otherwise we defined it as an unmatched cluster, and the corresponding strain labels were annotated using a varying color palette to identify these tips (see example in Figures 2A, 2B). We defined accuracy as the proportion of matched clusters to all PopPUNK/iterative-PopPUNK clusters. For example, for simulated data obtained with a recombination rate (r=0.001) and mutation rate (μ=0.001, Figure 2A, 2B), PopPUNK and iterative-PopPUNK (after step 4) generated 9 and 69 clusters, with the accuracy of 100% (9/9) and 99% (68/69), respectively. Our accuracy estimate is typically large, because strains from unmatched clusters were closely related in the simulated tree, although not fully clustered together (Figure 2B).

Overall, iterative-PopPUNK generates many more (on average ~6-fold greater) clusters than PopPUNK with similar accuracy across a range of different mutation and recombination rates(Figure 2C, 2D). A universal model we can recommend for iterative-PopPUNK is GMM with two components (K = 2), because more datasets could be analysed using this setting (DBSCAN fits sometimes fail to converge). Under this model, the average accuracy of clusters remains high across parameter choices (~95%). Therefore, we set it as the default model of iterative-PopPUNK and applied it in further analysis.

We next tested the performance of iterative-PopPUNK under different relative genetic similarity ($P_{max\text{-}cluster\text{-}dist}$) cutoffs. We again use the results obtained for simulated data under a mutation and recombination rate (μ=0.001, r=0.001, Figure 3A) as an example, and mapped iterative-PopPUNK clusters under different cutoffs back to simulated trees. As expected, the genetic diversity of strains within a cluster increases as the cutoff increases. Overall, the number of clusters obtained first increases, and then decreases with cutoff point, except in the high recombination rate scenario (Figure 3B). For each dataset we analyzed, the accuracy of clusters across similarity cutoffs was similar. However, the accuracy between different datasets varies, with higher average accuracy of clusters in datasets with high recombination rates.

Application of iterative-PopPUNK to real data shows consistency with phylogenetic results

We applied iterative-PopPUNK with default settings to analyze the genomic datasets of eight bacterial species with different mutation and recombination rates (Table S1) and obtained the cluster assignments under six relative similarity cutoffs ($P_{max\text{-}cluster\text{-}dist}$: 1%-99%). In addition, we performed core genome alignments for these datasets, and constructed maximum-likelihood (ML) phylogenetic trees based on core-genome SNPs. To measure the consistency between iterative-PopPUNK and alignment-based method, we mapped the clusters obtained under different cutoffs to the corresponding ML trees, and calculated matching accuracy (i.e. the proportion of matched clusters to all clusters) in the same manner as described above.

We collated the dataset for *Escherichia coli* by subsampling 500 genomes from 50 largest lineages (10 per lineage), which together represent the entire genomic diversity of human-

isolated *E. coli* (Horesh et al. 2021). The entire, fully sampled, dataset has previously been analyzed by PopPUNK, and these 50 lineages were defined based on PopPUNK clusters. High matching rates were identified, with the overall rate of 75%. When a small cutoff ($P_{max\text{-}cluster\text{-}dist}$=7.9%) was applied, iterative-PopPUNK estimated 44 clusters (singletons excluded; 94 clusters if singletons included), among which 43 of them could be perfectly matched with phylogenetic tree nodes. For this same dataset, 55 STs groups (singletons excluded; 92 STs if singletons included) could be identified, with 26 of them were exactly corresponding to tree nodes. Iterative-PopPUNK clusters roughly corresponded to samples from a single sequence type (matching rate: 27% [15/56]; Fig5). Using a medium cutoff ($P_{max\text{-}cluster\text{-}dist}$ =20%), 42 clusters were being estimated by iterative-PopPUNK which exactly corresponded to 42 clusters from phylogenetic method using 20% SNPs distance as cutoff. Under a large cutoff ($P_{max\text{-}cluster\text{-}dist}$=50%), iterative-PopPUNK cluster closely corresponded to strains from a phylogroup. Of these 9 obtained clusters, Iterative-PopPUNK failed to find phylogroup C, which appears to be a sub-clade of B1 according to the phylogenetic tree and so does not deserve a phylogroup label according to current data. Six perfectly corresponded to phylogroups (A, B2,E, F *Shigella flexneri and Shigella sonnei*) and the other two corresponded to the sub-clades of phylogroups D. This example highlighted the advantage of iterative-PopPUNK, which allows multi-level clustering in a unified way and users can select their "level" (e.g. sequence type or phylogroup level) according to their analytic purpose.

The datasets of the other six species were assembled by simply combining publicly available genomes in NCBI RefSeq database. Iterative-PopPUNK did not produce useful results for two species with low genetic diversify, namely *Mycobacterium tuberculosis* and *Bacillus anthracis*. For the other four species, the average matching rates were 93%, and higher rates were identified in species with high recombination rates such as *Helicobater pylori* and *Vibrio parahaemolyticus*. The matching rates at different similarity cutoffs were generally similar (Figure 5).

Discussion

Over the last two decades, with the development of genome sequencing technology and the rapid decline of costs, genome data is accumulating at an unprecedented speed. For some common pathogens such as *Salmonella enterica* and *Escherichia coli*, there are now more than 200,000 genomes in public databases such as Enterobase and Genometracker (Brown et al. 2021). Analyzing large-scale genome datasets is still challenging, and alignment-free methods such as PopPUNK are one of the best solutions to address this. PopPUNK was initially designed to identify a single level of clusters by optimizing a heuristic, but in practice different cluster resolutions are required, depending on the purpose. To solve this problem, we developed iterative-PopPUNK, which allows estimation of a partially resolved genealogical tree and enables clustering at different levels of sequence similarity. Iterative-PopPUNK inherits the advantages of PopPUNK, which is fast and consumes few computational resources, and it is easily extendable when new data needs to be added. On this basis, Iterative-PopPUNK provides greater flexibility and allows the resolution of the clustering to be adjusted according to the specific purpose.

Currently, the most used method for establishing the relationships between bacterial strains is phylogenetic analysis inferred from genomic alignments. This method usually requires a

reference genome, and inappropriate reference selection may introduce bias. This approach is also not typically scaleable to large-scale genome datasets (e.g. >10000) and runtimes can become prohibitively long, even with generously provisioned computational resources. For example, on a high-end workstation a phylogenomic analysis of ~10,000 *Klebsiella pneumoniae* genomes took 14 days and required a maximum of ~500 Gb memory. Another kind of method for classifying strains is based on gene-based subtyping information, including the classicaly multi-locus sequence typing (MLST) which is based on several house-keeping genes, and more recently developed core-genome MLST (cg-MLST) or whole-genome MLST (wg-MLST). Compared with genome-wide variation such as SNPs, subtyping information is typically much more parsimonious at the loss of some resolution, which can be used to scale for the analysis of large-scale data. There have been methods for multi-level clustering of strains based on subtyping data. HierCC is one of the state-of-the-art pipelines for multi-level clustering analysis of large-scale dataset based on cgMLST. HierCC hierarchical clusters (HC) can be used for multi-level strain assignment. For example, in *Salmonella*, HC900 (<900 allele difference) and HC2500 (<2500 allele difference) represent strains from the same sequence type complex and species/subspecies, respectively. However, performing gene-by-gene subtyping-based clustering first requires an appropriate curated database of genes, and a stable strain nomenclature. These have not yet constructed for most bacterial species, especially for non-pathogenic species, and require ongoing maintenance, making this method unlikely to be the general solution for microbes.

The current version of iterative-PopPUNK has two main limitations. First, it does not perform well on species with low genetic diversity because boundaries tend to connect clusters together in a non-transitive manner. This can be solved by nearest neighbour approaches. Second, the range of the levels of clusters obtained depends on the model fit of within- and between-strains components. Sometimes, only a very limited number of levels can be obtained, especially when using the second type of relative similarity cutoff $P_{bet\text{-}mean\text{-}dist}$.

With the widely use of whole genome sequencing and rapid accumulation of genome sequences, large-scale genome dataset analysis may become normal in the near future. However, there is currently no ideal multi-level clustering method for large-scale dataset analysis. Iterative-PopPUNK is alignment-free, highly efficient even for very large-scale dataset, and does not reply on curated database and nomenclature, which can solve the limitations of the above methods, and we believe has the potential to be the general solution for multi-level clustering of many bacterial species.

Methods

Implementation of iterative-PopPUNK algorithm

Iterative-PopPUNK procedure is implemented in the 'poppunk_iterate' program installed as part of PopPUNK package. Source code is available on GitHub (https://github.com/johnlees/PopPUNK), and can be installed through conda, pip or manually. Users can run multi-level clustering procedure using the flags '--fit-model refine --multi-boundary' in the PopPUNK software. The subsequent QC procedure is implemented in the 'poppunk_iterate' program and uses output from multi-level clustering step as input.

Iterative-PopPUNK cluster accuracy evaluation using simulation data

We performed simulation using FastSimBac(De Maio and Wilson 2017) to test the accuracy of iterative-PopPUNK clusters. Fifteen parameter combinations were used in simulation to represent bacteria with different mutation and recombination rates (Table S1). Simulation outputs were converted to genome sequences in fasta format for further iterative-PopPUNK analysis using integrated simulator, *msformatter* and *ms2dna* (available from http://guanine.evolbio.mpg.de/bioBox), and the corresponding genealogical trees were extracted for comparison with iterative-PopPUNK clusters.

For the iterative-PopPUNK analysis of simulated genomes, we set the maximum accessory distance allowed to 1.0 to retain all isolates. In the first "model fitting" step, 2D Gaussian Mixture Models (2d GMM) from original PopPUNK was chosen to fit the distribution of core and accessory distances. Two components (--K 2), one for within-strain and the other one for between-strain were applied. The other parameters were set to default values (k=15-29 in steps of 4; sketch size 10^4), which have been shown to work well across most bacterial species.

Based on the nodes information of simulated genealogical tree, we assign simulated strains into tree-clusters. Each iterative-PopPUNK cluster was compared with all the tree-clusters using ete3 toolkit(Huerta-Cepas et al. 2016), respectively. If the strain assignment of an iterative-PopPUNK cluster is identical to one of these simulated tree-clusters, it is defined as a match iterative-PopPUNK cluster, and associated nodes were labeled using green/red faces alternatively in the simulated tree; otherwise it is unmatched and the leaves would be annotated using a certain color for each unmatched cluster. Accuracy is defined as the proportion of matched clusters to all iterative-PopPUNK clusters.

Application of iterative-PopPUNK in real data

The publicly available, assembled genomes of seven species were downloaded from the National Center for Biotechnology Information (NCBI) Reference sequence (RefSeq) database based on the taxonomy identifier (Table S2). Two steps of genome sequences QC were performed. First, we used CheckM(Parks et al. 2015) to assess the genome quality and only genome sequences with completeness >90% and contamination <5% were kept in further analysis. Second, we set 70% as whole genome coverage threshold to remove low-quality genomes or outlier samples that may not be part of the species. After QC, the same iterative-PopPUNK pipeline used for simulation data were applied to real data analysis. The 500 *E. coli* genomes dataset were collected from Horesh G, et al. Microb Genom. 2021.

QC-passed genome sequences were further aligned to the reference genome of each species (Table S2) using MUMmer (Kurtz et al. 2004) to generate whole genome alignments. Core-genome (regions present in >99% isolates) SNPs were called using snp-sites(Page et al. 2016) based on the alignments. Maximum-likelihood (ML) phylogenetic tree were constructed using FastTree(Price et al. 2010) based on core-genome SNPs. We adopted the method used in Iterative-PopPUNK cluster accuracy evaluation to calculate the matching rate of iterative-PopPUNK clusters and ML trees, by replacing the simulated trees to ML trees.

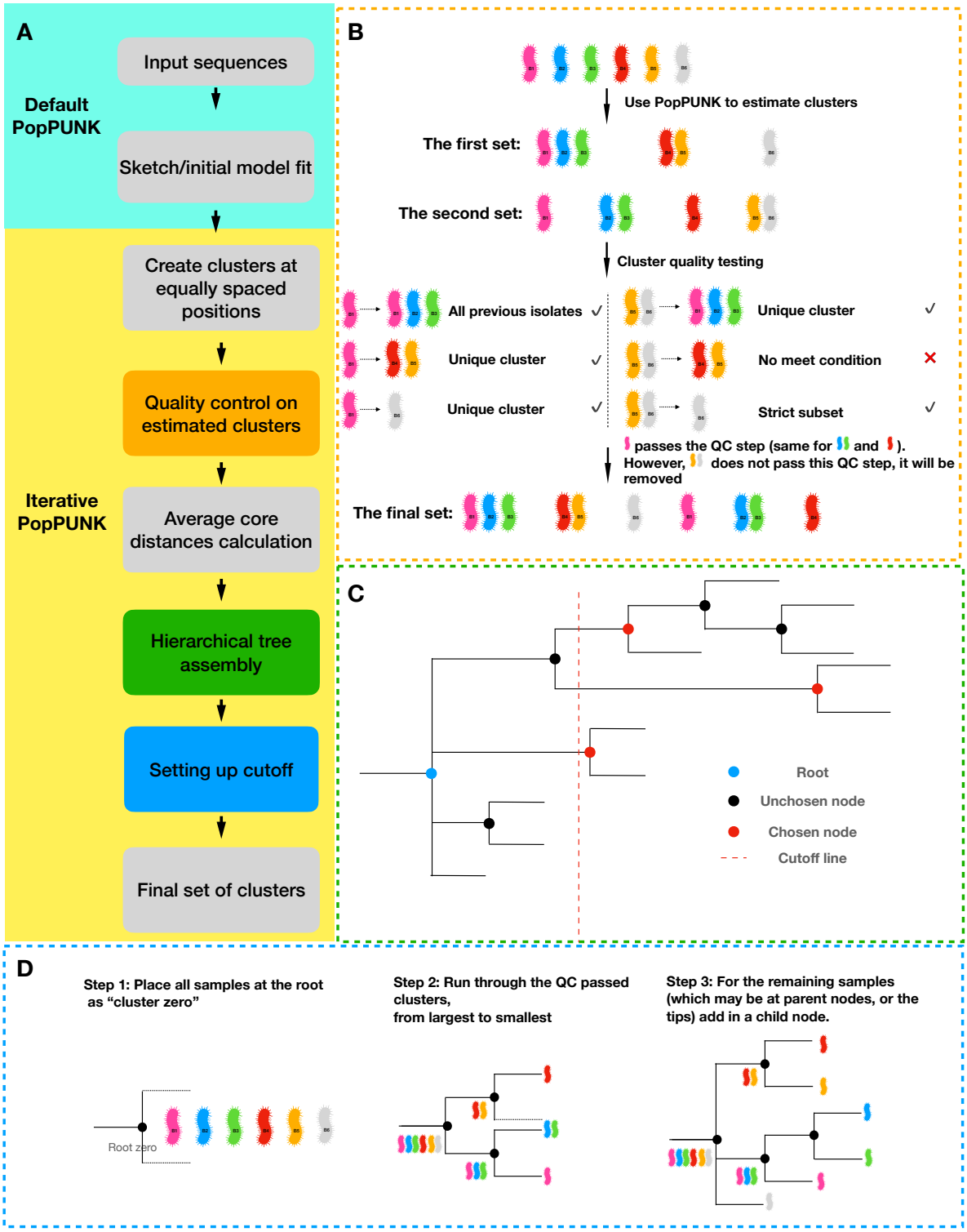

Figure 1: Workflow for developing iterative-PopPUNK. A). Steps for designing iterative-PopPUNK. After inputing sequences data, PopPUNK creates local pairwaise distances database which was furthur used to estimate clusters. QC steps for removing inconsistent clusters was showin in orange box. Green box shows how to nest these obained clusters into an iterative-PopPUNK tree. Methods for choosing final set of clusters were presented in blue box; B). QC algorithm. Three conditions are discribed in detals in methods, which are: 1) the new clusters contain all of isolates from previous clusters; 2). the new cluster is unique and contains none of items from previous clusters; 3). the new cluster is a strict subset of one of the previous clusters. C). Algorithm for cutting iterative-PopPUNK tree. The goal is to choose the closest node to the cutoff line but with a smaller value. D). Hierachical tree assembly. The dashed lines indicated these potential branches during the tree assembly process.

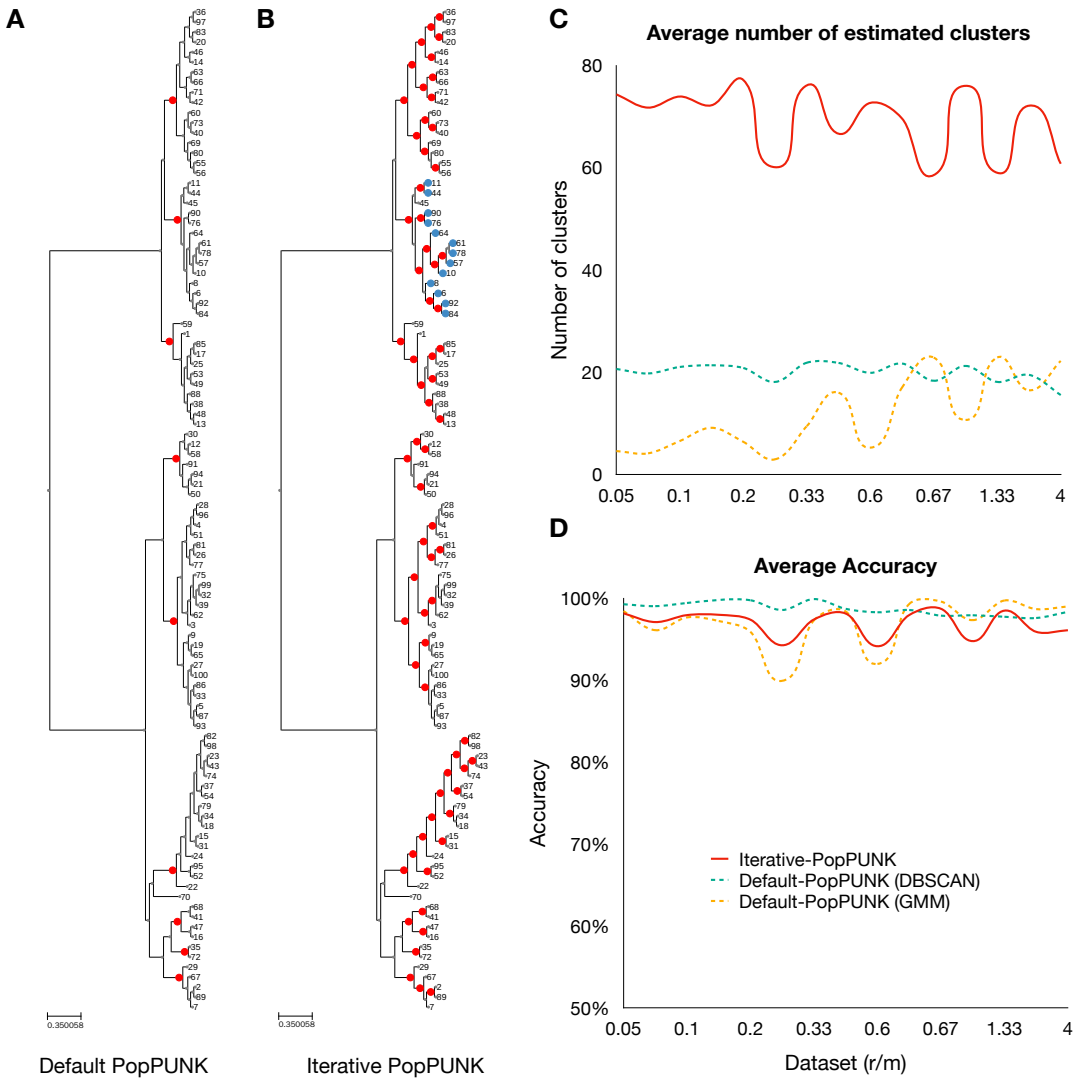

Figure 2: The performance of iterative-PopPUNK. A1). A genealogical tree with 9 nodes (clusters) estimated by default PopPUNK; A2). The same genealogical tree with 69 clusters estimated by iterative-PopPUNK, among which 68 can be totally matched with genealogical tree nodes while 1 of them cannot be matched and these isolates (leaves) were being annotated using different colors. B1). A line graph shows the average number of estimated clusters by PopPUNK and by iterative-PopPUNK. The solid lines present the results for iterative-PopPUNK, and the dash lines are for default-PopPUNK. B2). The accuracy results for both default PopPUNK and iterative-PopPUNK. The accuracy is calculated from total number of matched nodes divided by total number of estimated clusters. The chosen example dataset has a recombination rate r=0.001 and mutation rate =0.001

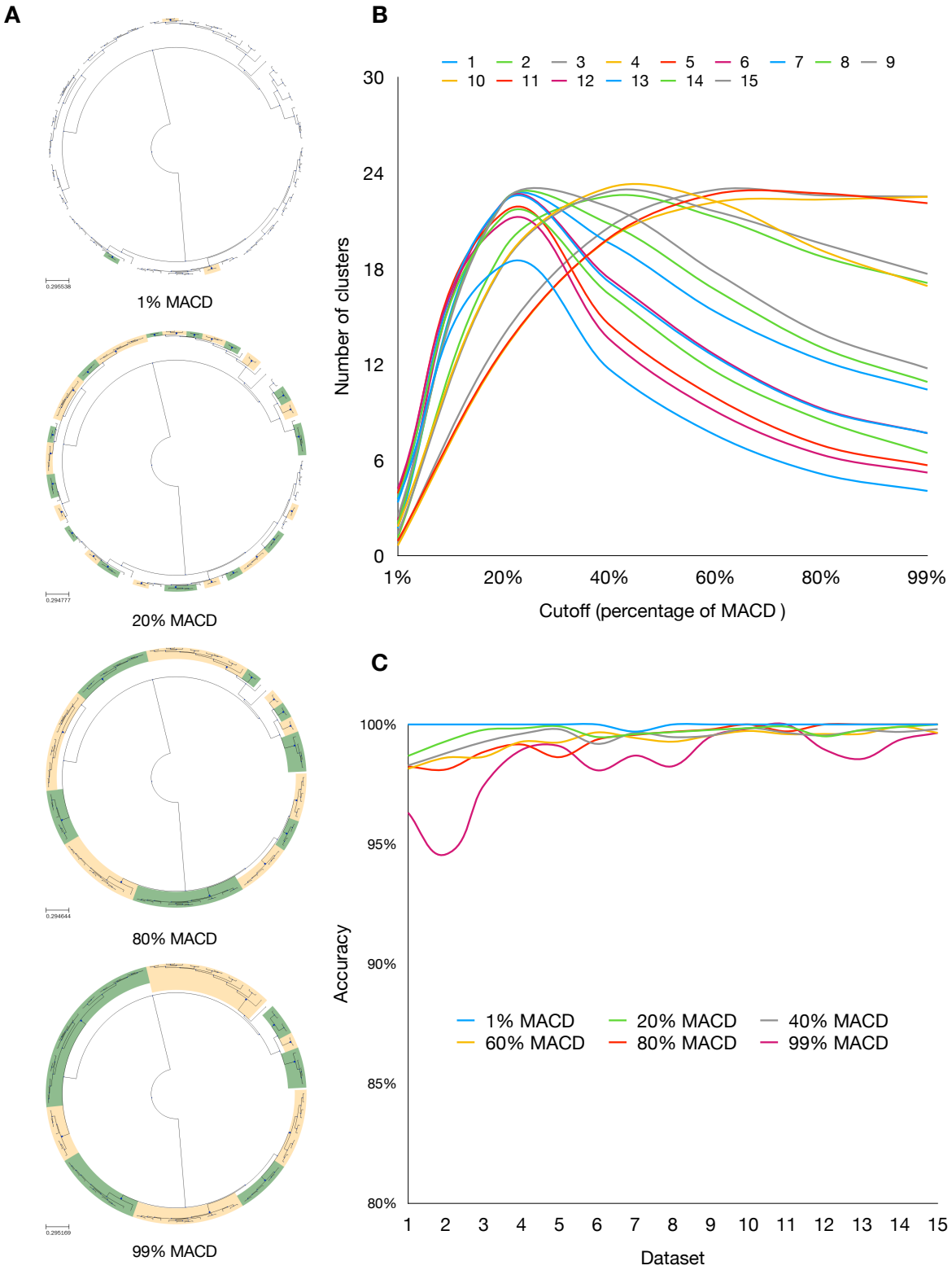

Figure 3: The performance of iterative-PopPUNK at multiple levels. A) the comparison of iterative-PopPUNK clusters with the genealogical tree (from 100 samples) at four levels: 1%, 20%, 80%, and 99% of the MACD. These green and orange faces represent matched clusters with tree nodes respectively. For these unmatched clusters, the tree leaves are annotated using different colors. B). 1). The average number of clusters while moving the cutoff lines in iterative-PopPUNK tree. The value is calculated by the total number of estimated clusters divided by total number of genomes at each level; 2). The accuracy of estimated clusters at each level. The accuracy is calculated by the total number of matched clusters (with tree nodes) divided by the total number of estimated clusters. The chosen example dataset has a recombination rate r=0.006 and mutation rate μ=0.01

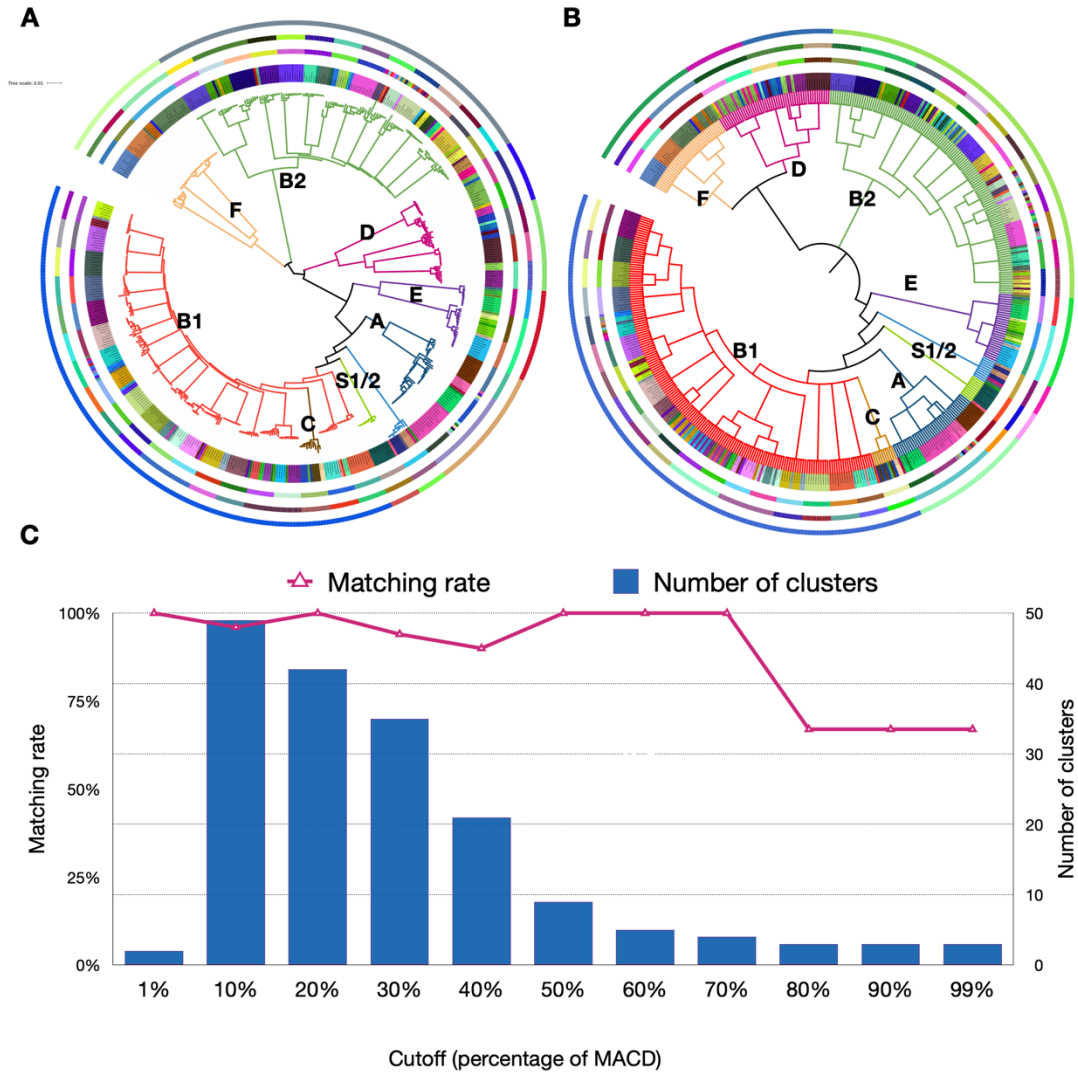

Figure 4: Population structure of E.coli at different iterative-PopPUNK levels. A). Comparison of PopPUNK clusters and phylogenetic tree. B). Comparison of PopPUNK clusters and Iterative-PopPUNK tree. The outer solid ring represents 9 PopPUNK clusters at 50% of the MACD, corresponding to 9 E.coli phylogroups. The middle ring contains 42 clusters at 20% of the MACD, corresponding to 42 clusters from phylogenetical tree at 20% SNP distance. The inner solid ring shows 94 PopPUNK clusters at 7.9% of the MACD, corresponding to 92 E.coli STs groups. Colors in the tree leaves and branches are used to present these 92 STs and 9 phylogroups respectively. The parameters used for estimating PopPUNK clusters were all set to default with 2D GMM model (two components). C). the distribution of matching rate and the number of clusters being estimated at multiple levels for this set of E.coli genomes. The tree was plotted using iTOL (https://itol.embl.de/).

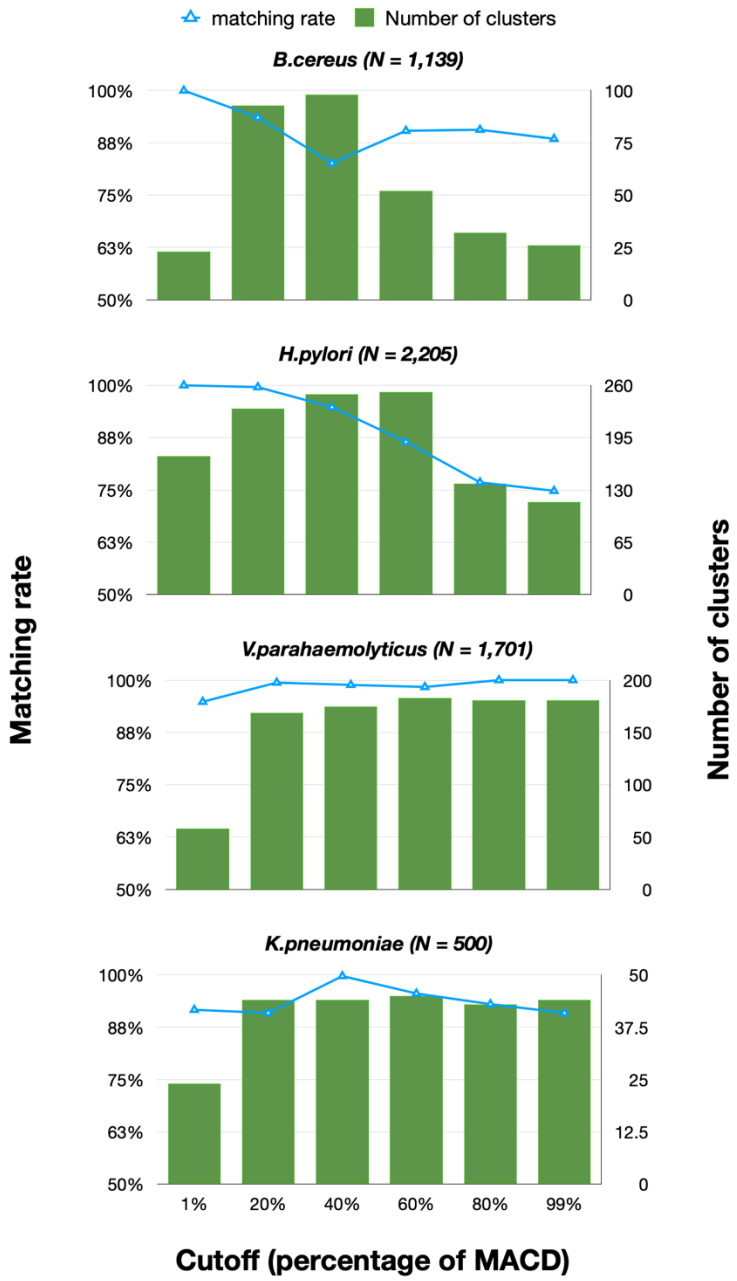

Figure 5: The distribution of matching rate and the number of clusters being estimated at multiple levels. the cluster is defined as a group with two or more strains (singletons are not considered in counting the number of clusters)

References

Brinda K, Callendrello A, Ma KC, MacFadden DR, Charalampous T, Lee RS, Cowley L, Wadsworth CB, Grad YH, Kucherov G et al. 2020. Rapid inference of antibiotic resistance and susceptibility by genomic neighbour typing. *Nat Microbiol* **5**: 455–464.

Brown B, Allard M, Bazaco MC, Blankenship J, Minor T. 2021. An economic evaluation of the Whole Genome Sequencing source tracking program in the U.S. *PLoS One* **16**: e0258262.

Croucher NJ, Didelot X. 2015. The application of genomics to tracing bacterial pathogen transmission. *Curr Opin Microbiol* **23**: 62–67.

De Maio N, Wilson DJ. 2017. The Bacterial Sequential Markov Coalescent. *Genetics* **206**: 333–343.

Falush D. 2016. Bacterial genomics: Microbial GWAS coming of age. *Nat Microbiol* **1**: 16059.

Gardy JL, Loman NJ. 2018. Towards a genomics-informed, real-time, global pathogen surveillance system. *Nat Rev Genet* **19**: 9–20.

Gu W, Miller S, Chiu CY. 2019. Clinical Metagenomic Next-Generation Sequencing for Pathogen Detection. *Annu Rev Pathol* **14**: 319–338.

Horesh G, Blackwell GA, Tonkin-Hill G, Corander J, Heinz E, Thomson NR. 2021. A comprehensive and high-quality collection of Escherichia coli genomes and their genes. *Microb Genom* **7**.

Huerta-Cepas J, Serra F, Bork P. 2016. ETE 3: Reconstruction, Analysis, and Visualization of Phylogenomic Data. *Mol Biol Evol* **33**: 1635–1638.

Kurtz S, Phillippy A, Delcher AL, Smoot M, Shumway M, Antonescu C, Salzberg SL. 2004. Versatile and open software for comparing large genomes. *Genome Biol* **5**: R12.

Lees JA, Harris SR, Tonkin-Hill G, Gladstone RA, Lo SW, Weiser JN, Corander J, Bentley SD, Croucher NJ. 2019. Fast and flexible bacterial genomic epidemiology with PopPUNK. *Genome Res* **29**: 304–316.

Page AJ, Taylor B, Delaney AJ, Soares J, Seemann T, Keane JA, Harris SR. 2016. SNP-sites: rapid efficient extraction of SNPs from multi-FASTA alignments. *Microb Genom* **2**: e000056.

Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW. 2015. CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res* **25**: 1043–1055.

Power RA, Parkhill J, de Oliveira T. 2017. Microbial genome-wide association studies: lessons from human GWAS. *Nat Rev Genet* **18**: 41–50.

Price MN, Dehal PS, Arkin AP. 2010. FastTree 2--approximately maximum-likelihood trees for large alignments. *PLoS One* **5**: e9490.

Sintchenko V, Holmes EC. 2015. The role of pathogen genomics in assessing disease transmission. *BMJ* **350**: h1314.

Tonkin-Hill G, Lees JA, Bentley SD, Frost SDW, Corander J. 2019. Fast hierarchical Bayesian analysis of population structure. *Nucleic Acids Res* **47**: 5539–5549.

Zhou Z, Charlesworth J, Achtman M. 2021. HierCC: A multi-level clustering scheme for population assignments based on core genome MLST. *Bioinformatics* doi:10.1093/bioinformatics/btab234.