PhyDBSCAN2: Phylogenetic Tree Density-Based Spatial Clustering of Applications With Noise and Automatically Estimated Hyperparameters

Lida Hooshyar[1], *Ryan* Godin[1], *Anna* Artiges[1], and *Nadia* Tahiri[1*]

[1]Department of Computer Science, University of Sherbrooke, 2500, boulevard de l'Université, J1K 2R1, QC, Canada

Abstract.
Phylogenetic analyses often generate numerous tree topologies, creating conflicts that require resolution through consensus strategies. Conventional single-tree consensus methods have inherent limitations, as they do not capture topological diversity and are sensitive to outliers. This study presents a novel approach, PhyDBSCAN, that applies the density-based spatial clustering of applications with noise (DBSCAN) algorithm to ensembles of phylogenetic trees. The refined DBSCAN method includes an optimized, data-driven procedure for estimating the hyperparameters *epsilon* and $MinPts$, developed specifically for the Robinson-Foulds (RF) distance. This approach clusters trees, partitioning them into a single cluster for homogeneous data and multiple clusters for heterogeneous data, preserving topological diversity and enhancing consensus construction. PhyDBSCAN has a time complexity of $\mathcal{O}(nN^2)$, where n is the number of leaves and N is the number of phylogenetic trees. The efficiency of the new method was assessed using real data comprising 35 genes from 43 methanogen species.

Keywords: Phylogenetic analysis, Consensus tree, PhyDBSCAN clustering, and Robinson–Foulds (RF) distance.

1 Introduction

The evolution of species is commonly represented by a phylogenetic tree that depicts the evolutionary history of a group of species. These trees are usually constructed from nucleotide sequences or morphological characters [1–6]. It is possible to generate a collection of trees derived from the same or different datasets. To identify a tree that is representative of this collection, various approaches [7–10] have been developed, depending on whether the trees are defined on the same set of leaves or overlapping sets of leaves. In the context where all trees share the same set of leaves, consensus tree methods [11–13] are applied to summarize the most common evolutionary relationships, while if the trees have overlapping but not identical sets

*e-mail: nadia.tahiri@usherbrooke.ca

of leaves, supertree methods [14–16] are used to merge them into a single comprehensive tree. In particular, a consensus tree provides a computational algorithm that synthesizes multiple phylogenetic trees into a single representative topology, thereby capturing the most frequently occurring evolutionary relationships and highlighting patterns consistently supported across analyses. This method allows for the detection of shared evolutionary patterns and relationships across multiple phylogenetic analyses [17, 18].

A method for producing consensus trees is the strict consensus tree [19], which retains only those bipartitions present in all input trees. While this approach ensures that only universally supported relationships are included, a key limitation is that it may discard bipartitions that are well-supported in most, but not all, trees, leading to highly unresolved topologies and loss of informative phylogenetic signal. In comparison, the majority consensus tree [19] incorporates all bipartitions present in more than half of the input trees, typically producing a partially resolved tree. However, this method may still fail to capture bipartitions that, although strongly supported in a substantial minority of input trees, occur in fewer than half of them, resulting in partial loss of phylogenetic signal [20, 21]. Nevertheless, it behaves more permissively than the strict consensus by allowing greater resolution. The extended majority consensus tree further improves upon this approach by not only including the majority bipartitions but also sequentially adding additional compatible bipartitions in decreasing order of support [22]. This strategy retains more phylogenetic information from less frequent yet compatible bipartitions, often yielding trees that are nearly fully resolved and better reflect the underlying evolutionary relationships among taxa. The Nelson consensus [23] constructs a tree by including compatible bipartitions with the highest total weight. However, this approach requires solving a maximum weight clique problem on a compatibility graph, which makes it computationally demanding, especially for large sets of input trees. These methods represent only a subset of the broader class of consensus approaches, which also includes variants such as greedy consensus, Adams consensus, and frequency-difference consensus, each designed to balance resolution and accuracy under different phylogenetic conditions.

The problem with these methods is that, while numerous consensus tree approaches exist, they typically generate a single tree. This can lead to information loss and susceptibility to outliers [24, 25]. In particular, traditional consensus methods often force all candidate trees into a single topology and may overlook existing topological diversity. Many phylogenetic inference methods can generate multiple candidate trees for a given dataset. As an example, the Maximum Parsimony (MP) method [26] identifies binary trees with the lowest parsimony score, defined as the minimum tree length, i.e., the total number of character changes across all branches. Depending on the dataset, MP analysis can produce thousands of equally parsimonious trees, highlighting the challenge of summarizing phylogenetic information into a single consensus tree. One widely recognized method is the construction of multiple consensus trees, which allows for a more broader and more informative representation of the underlying phylogenetic signal [6, 27–29]. Nevertheless, even within these different approaches, difficulties persist, as the input trees cannot be merged into a single tree. Therefore, a preliminary step of clustering the phylogenetic trees is essential. Specifically, given a collection of N trees defined in n species, one must identify an optimal partition of trees that share similar evolutionary patterns while simultaneously accounting for potential outliers. This task is non-trivial, as inappropriate partitioning can obscure meaningful biological relationships or amplify noise in the data.

To overcome this limitation, we propose an alternative postprocessing strategy based on the density-based spatial clustering of applications with noise (DBSCAN) algorithm [30], termed Phylogenetic DBSCAN (PhyDBSCAN). This method applies a density-based clustering framework to group phylogenetic trees into coherent subsets, with each cluster subsequently summarized by its corresponding consensus tree. This method handles outliers more effectively than commonly used methods such as k-means and k-medoids [31].

This study makes the following contributions: First, an adaptation of DBSCAN to phylogenetic trees using Robinson-Foulds (RF) [32] distance as the basis for clustering is presented. Second, a data-driven procedure for determining the parameters *epsilon* and $MinPts$ is proposed, avoiding manual tuning; these parameters are not specified initially but are optimized automatically. Third, by clustering trees, PhyDBSCAN identifies a consensus tree for each homogeneous group, thereby preserving topological diversity. Finally, noisy or inconsistent trees are isolated, ensuring that the resulting consensus trees reflect reliable phylogenetic signals. Various theoretical optimization problems associated with these outputs are addressed, and preliminary progress is presented and discussed in the section dedicated to clustering criteria.

This paper is organized as follows. Section 2 introduces the foundational definitions and key concepts underlying this study. Section 3 describes the methodology used to determine the optimal neighborhood radius and minimum cluster size parameters for DBSCAN clustering. Section 4 presents the evaluation of the PhyDBSCAN algorithm using 43 methanogen genomes associated with Posidonia oceanica. Finally, Section 5 summarizes the main results, discusses their implications, and suggests directions for future work.

2 Definitions and notation

Definition 1 *(Phylogenetic tree) A phylogenetic tree* $T = \{V(T), E(T)\}$ *is a directed, acyclic, and unweighted graph, where* $V(T)$ *denotes the set of vertices (nodes) of* T *and* $E(T)$ *denotes the set of edges of* T*. Each internal vertex has at least two children, and each leaf is uniquely labeled. For any two vertices* $u, v \in V(T)$*, an edge* $(u, v) \in E(T)$ *represents a directed link from* u *to* v*, indicating a parent–child relationship where* u *is the parent and* v *is the child.*

Definition 2 *(Consensus tree) Let* $\mathcal{T} = \{T_1, \ldots, T_N\}$ *be a set of* N *phylogenetic trees on the same set of species* $L = L(T_1) = \cdots = L(T_N)$*. A consensus tree is a phylogenetic tree noted as* T_c *with* $L(T_c) = L$ *that summarizes all* N *input trees.*

A leaf-labeled tree topology can be decomposed into a set of bipartitions using the following method. Every edge, upon removal from the tree, induces a bipartition of the leaves. Consequently, each edge can be associated with its induced bipartition.

Definition 3 (Robinson–Foulds distance) *Let* T_1 *and* T_2 *be two unrooted phylogenetic trees on the same set of leaves denoted by* L *where* $|L| = n \geq 3$*. Let* $E(T_1)$ *and* $E(T_2)$ *denote the sets of bipartitions of* L *induced by the internal edges of* T_1 *and* T_2*, respectively. The Robinson–Foulds distance between* T_1 *and* T_2 *is defined as* $d_{\mathrm{RF}}(T_1, T_2) = |E(T_1) \setminus E(T_2)| + |E(T_2) \setminus E(T_1)|$.

3 Methods

The objective of this research is to evaluate the potential of using DBSCAN to classify a set of phylogenetic trees based on their RF distance properties. One of the advantages of using DBSCAN to cluster trees with RF distance is that the RF distance can be transformed into a quasi-hypermetric distance by taking its square root, which satisfies the triangle inequality [33]. However, it may not satisfy the condition of a metric distance, which require the four-point property [27]. The RF distance between two trees defined on n leaves has a time complexity of $\mathcal{O}(n)$ [34] and consequently $\mathcal{O}(nN^2)$ between N trees defined on n leaves. The DBSCAN algorithm is known to be effective in clustering Euclidean space data. This algorithm has a time complexity of $\mathcal{O}(N \log N)$ for the clustering step when appropriate indexing structures are used, which can be more efficient than the naive $\mathcal{O}(N^2)$ implementation [35]. This efficiency is particularly crucial for large datasets in a phylogenetic context, where the number of species can be significant.

3.1 DBSCAN in Euclidean space

The DBSCAN algorithm, originally introduced by Ester et al., is a commonly used clustering algorithm in scientific literature and has also been applied in recent works such as [36]. DBSCAN is designed to discover arbitrary-shaped clusters in any dataset X and detect noise points. It partitions data according to point density, determined by the labels assigned to each point based on two hyperparameters: the radius value *epsilon* and density threshold $MinPts$ value, which measures the distance between two points based on the distance function. Some concepts and terms to explain the DBSCAN algorithm can be defined as follows.

Definition 4 *(epsilon-neighbor$_x$) The epsilon-neighbor of a point $x \in X$ (denoted as epsilon-neighbor$_x$) is a set of points inside an epsilon radius around x, such that epsilon-neighbor$_x$ $= \{y \in X | dist(x, y) \leq epsilon\}$.*

Definition 5 *(Point property) An observation x is classified as one of the following classes:*

- *Core point, denoted as core$_x$, if and only if point x satisfies the condition $|$epsilon-neighbor$_x| \geq MinPts$;*
- *Border point, denoted as border$_x$, if and only if point x satisfies the condition $|$epsilon-neighbor$_x| < MinPts$ and there exists at least one epsilon-neighbor$_x$ that is a core$_x$;*
- *Noise point, denoted as noise$_x$, otherwise.*

3.2 Optimal estimation of the *epsilon* value

The provided Equation 1 determines an optimal *epsilon* value for a given dataset of pairwise distances among a set of trees. The main objective is to identify the elbow points in the sorted distances for each tree and calculate the average of these elbow values to obtain the best *epsilon* value.

$$epsilon = \frac{1}{N} \sum_{i=1}^{N} \left(\max_{j=1}^{N-1} |\mathrm{RF}(T_i, T_j) - \mathrm{RF}(T_i, T_{j+1})| \right), \tag{1}$$

where N is the number of phylogenetic trees, $\mathrm{RF}(T_i, T_j)$ and $\mathrm{RF}(T_i, T_{j+1})$ are the RF distances between phylogenetic trees T_i and T_j, and between T_i and T_{j+1}, respectively. For each tree T_i, the quantity of interest is the maximum absolute difference between consecutive distances in its sorted RF distance vector. The computation of the parameter *epsilon* is performed in quadratic time with respect to N, that is, in $\mathcal{O}(N^2)$.

In the initial step of the DBSCAN algorithm, RF distances are computed between all phylogenetic trees in the set. Subsequently, for each phylogenetic tree i, the critical value of $epsilon_i$ is determined by evaluating the distance to its first neighbor before the elbow. Equation 1 outlines this process. It iteratively identifies elbow points for each tree by calculating the absolute differences between consecutive distances. The critical *epsilon* values for all phylogenetic trees are then obtained by averaging the distances over all nearest neighbor distances. This adaptive approach to determining the optimal *epsilon* value for clustering aligns with the intrinsic structure of the dataset. The RF distances and elbow points contribute to the geometrically informed assessment, increasing the robustness and adaptability of the clustering algorithm.

3.3 Optimal estimation of the *MinPts* value

Let $\mathcal{T} = \{T_1, T_2, \ldots, T_N\}$ be a set of N trees in the RF metric space with distance function $\mathrm{RF}(\cdot, \cdot)$. The parameter *MinPts* is defined as follows:

$$MinPts = \left\lfloor \frac{1}{N} \left(\sum_{i=1}^{N} \mathrm{cardinality}(T_i, epsilon) \right) \right\rfloor, \tag{2}$$

where

$$\mathrm{cardinality}(T_i, epsilon) = \sum_{j=1}^{N} \begin{cases} 1 & \text{if } \mathrm{RF}(T_i, T_j) < epsilon, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathrm{cardinality}(T_i, epsilon)$ represents the number of the phylogenetic trees within the *epsilon* distance from phylogenetic tree T_i. The computation of the parameter $MinPts$ requires quadratic time with respect to N, that is, $\mathcal{O}(N^2)$.

In Equation 2, the number of neighbors for a given tree is counted, if the distance between a tree T_i and T_j is less or equal to the *epsilon* value determined in Equation 1 the two trees are considered neighbors. Once the cardinalities are obtained, the average is calculated and rounded down to ensure consistency within the same cluster. In addition, Equation 3 is used to systematically determine the number of neighbors within the *epsilon* distance from the phylogenetic tree T_i.

The PhyDBSCAN algorithm described in Algorithm 1 was implemented in C++, and its complete source code is publicly available without restrictions at https://github.com/tahiri-lab/PhyDBSCAN.

3.4 Properties

The following theorems represent several properties of the PhyDBSCAN algorithm.

Algorithm 1 PhyDBSCAN

Require:
$\mathcal{T}$: N unrooted binary phylogenetic trees
Ensure:
optimalPartition: K optimal partitioning of N trees
bestEpsilon: best *epsilon* value
bestMinPts: best $MinPts$ value
function PyDBSCAN($\mathcal{T}$)
  bestEpsilon ← CalculateEpsilon(RF) ▷ Eq. 1
  bestMinPts ← CalculateMinPts(RF, *bestEpsilon*) ▷ Eq. 2 and 3
  optimalPartition ← DBSCAN(RF, *bestEpsilon*, *bestMinPts*)
  return *optimalPartition*
end function

3.4.1 Time complexity

Theorem 1 *Let $\mathcal{T} = \{T_1, T_2, \ldots, T_N\}$ be a set of N trees defined on n species. The time complexity of the PhyDBSCAN algorithm for identifying a clustering of $\mathcal{T}$ is $\mathcal{O}(nN^2)$.*

Proof 1 *The PhyDBSCAN algorithm consists of four stages. Pairwise RF distances are computed in $\mathcal{O}(nN^2)$ time, where n is the number of taxa per tree and N the number of trees. The parameters epsilon and $MinPts$ are then obtained, each in $\mathcal{O}(N^2)$ time, followed by the DBSCAN clustering step with complexity $\mathcal{O}(N \log N)$. The overall time complexity is $\mathcal{O}(nN^2)$.*

3.4.2 Correctness

Theorem 2 *The PhyDBSCAN algorithm does not always return the correct clustering, where a clustering is considered correct if it partitions $\mathcal{T} = \{T_1, T_2, \ldots, T_N\}$ into the set of density-connected components under the RF metric with parameters epsilon and $MinPts$.*

Proof 2 *A counterexample can be constructed using 9 rooted trees, denoted by $\mathcal{T} = \{T_i \mid 1 \leq i \leq 9\}$, defined on the same set of 12 leaves $V(\mathcal{T}) = \{A, B, C, D, E, F, G, H, M, N, O, P\}$ (see Figure A5). The trees are grouped topologically into three distinct sets: $\mathcal{T}_1 = \{T_1, T_2, T_3\}$, $\mathcal{T}_2 = \{T_4, T_5, T_6\}$, and $\mathcal{T}_3 = \{T_7, T_8, T_9\}$. Within each cluster, the trees differ by a permutation of two leaves. Moreover, the trees in $\mathcal{T}_1$ and $\mathcal{T}_2$ share the same sub-tree defined on the leaves $\{M, N, O, P\}$ so these two clusters are similar compared to $\mathcal{T}_3$. Table A1 presents the pairwise matrix of RF distances normalized between 0 and 1.*

In this counterexample, the value of ϵ calculated by PhyDBSCAN is 0.555, and the corresponding $MinPts$ value is 4. The PhyDBSCAN algorithm identifies a cluster consisting of the trees in $\mathcal{T}_1$ and $\mathcal{T}_2$ based on these parameters. The trees in $\mathcal{T}_3$ are labeled as noise points, as none of them have four neighbors within a distance less than or equal to epsilon. In this case, PhyDBSCAN identifies only one cluster and incorrectly labels the other cluster as noise.

As shown in Proof 2, PhyDBSCAN has the potential to incorrectly label clusters as noise. This occurs because the outlier sets differ in their topology by some RF

measure, such as, in cases where a sets topology is completely different from the others. However, identifying the exact value and its conditions is outside the scope of this study.

4 Results

This study examines the genomic data from methanogens, strictly anaerobic archaea that derive energy from metabolizing simple carbon compounds (e.g., CO_2, CO, acetate, formate, methyl groups) and hydrogen, resulting in methane production [37]. Specifically, a focus on *P. oceanica* in the Mediterranean Sea, examining the intricate mechanisms underlying methane emissions from sediments beneath both living and dead seagrass, as well as from the root growth meadows of *P. oceanica*. The primary objective of this study is to identify the evolutionary groups responsible for methane production. The phylogeny of methanogens is diverse, classified into two classes and six major orders, represented by different colors in this article. Class I comprises *Methanopyrales* (black), *Methanobacteriales* (blue), and *Methanococcales* (purple), while Class II includes *Methanomicrobiales* (green), *Methanocellales* (red), and *Methanosarcinales* (orange). The dataset was selected due to uncertainties in the relationships between orders within each class and among species within each order.

Genomic sequences were processed following a meticulous protocol. Initially, the study identifies 43 reference genomes of methanogens on the National Center for Biotechnology Information (NCBI), representing all six major orders. The preprocessing of data involves compiling a list of known protein genes in the 43 genomes, identifying 35 genes present in 32 to 43 genomes, each occurring once per genome. This preprocessing is completed using a Python (version 3.10) script and is accessible on GitHub. Subsequently, the 35 gene alignments are prepared using Multiple Alignment using Fast Fourier Transform (MAFFT) (version 7) [38] and refined by the Block Mapping and Gathering with Entropy (BMGE) [39]. Phylogenetic trees are inferred using PhyML (version 3) [40].

Figure A1 illustrates the results obtained by applying the new PhyDBSCAN algorithm to the 35 genes of the 43 analyzed methanogens. In general, the three consensus trees mostly preserve the six methanogenic orders, except for the *Methanosarcinales*, *Methanocellales*, and *Methanomicrobiales* orders, which are particularly highlighted in Figures A2 and A3. The initial cluster (Figure A2) elucidates horizontal gene transfers (HGT) among three classes: *Methanosarcinales*, *Methanocellales*, and *Methanomicrobiales*, further corroborated by an alternative evolutionary trajectory observed within these three groups in the subsequent cluster (Figure A3). Lastly, the third cluster (Figure A4), underscores a distinct scenario aligning closely with the evolutionary history of the species, as compared to the reference tree, Figure A1.

5 Conclusion

In this article, a new algorithm is proposed for partitioning a set of phylogenetic trees into several clusters to infer multiple consensus trees. The DBSCAN algorithm is extended by incorporating the Robinson-Foulds topological distance within the framework of tree clustering. The new algorithm has the advantage of automatically adjusting the two hyperparameters, *epsilon* and $MinPts$, which are estimated by the PhyDBSCAN algorithm based on the data configuration. Using DBSCAN enables the detection of complex cluster shapes and the identification of outliers,

which is valuable for solving bioinformatics problems, such as identifying genes with similar evolutionary histories. In conclusion, the analysis emphasizes the results obtained by the PhyDBSCAN algorithm in clustering RF distance matrices and inferring consensus trees for methanogen data. A C++ program, named PhyDBSCAN, implementing the discussed tree partitioning algorithm is freely available on GitHub; https://github.com/tahiri-lab/PhyDBSCAN.

Multiple approaches may extend the scope of the study and improve its overall comprehensiveness: (1) adapting the methodology of supertrees (i.e., sets of phylogenetic tress defined on different, but overlapping leaf sets); (2) integrating advanced measures that account for branch lengths; (3) developing flexible and effective tuning of the *epsilon* and $MinPts$ parameters. However, validating simulation results is crucial for ensuring the reliability of subsequent analyses. The complexity of validating outcomes from the DBSCAN algorithm is rooted in two specific aspects: the optimal partitioning of data and the concomitant assignment of labels. While the selection of the Rand index appears as an appropriate choice for validating the optimal partition, the validation of labels remains a dimension inadequately addressed in the existing scientific literature [6, 27]. Future work aims to further refine the factor causing the PhyDBSCAN algorithm to produce inconsistent results on outliers and to formalize a more rigorous optimization of the new method through the proposal of a fixed algorithmic parameter.

References

[1] M.S. Lee, A. Palci, Morphological phylogenetics in the genomic age, Current Biology **25**, R922 (2015).
[2] O. Rieppel, Morphology and phylogeny, Journal of the History of Biology **53**, 217 (2020).
[3] A. Khastan, L. Hooshyar, A computational method to analyze the similarity of biological sequences under uncertainty, Iranian Journal of Fuzzy Systems **16**, 33 (2019).
[4] P. Forster, L. Forster, C. Renfrew, M. Forster, Phylogenetic network analysis of sars-cov-2 genomes, Proceedings of the National Academy of Sciences **117**, 9241 (2020).
[5] L. Hooshyar, M. Hernández-Jiménez, A. Khastan, M. Vasighi, An efficient and accurate approach to identify similarities between biological sequences using pair amino acid composition and physicochemical properties, Soft Computing **28**, 9341 (2024).
[6] N. Tahiri, M. Willems, V. Makarenkov, A new fast method for inferring multiple consensus trees using k-medoids, BMC evolutionary biology **18**, 1 (2018).
[7] D. Bryant, A classification of consensus methods for phylogenetics, DIMACS series in discrete mathematics and theoretical computer science **61**, 163 (2003).
[8] M. Wilkinson, J.L. Thorley, Efficiency of strict consensus trees, Systematic Biology **50**, 610 (2001).
[9] M. Wilkinson, Majority-rule reduced consensus trees and their use in bootstrapping., Molecular Biology and evolution **13**, 437 (1996).
[10] J. Jansson, C. Shen, W.K. Sung, Improved algorithms for constructing consensus trees, Journal of the ACM (JACM) **63**, 1 (2016).

[11] M.H.R. Sifat, N. Tahiri, A new algorithm for building comprehensive consensus tree, in *Graphs and more Complex structures for Learning and Reasoning: Proceedings of the 38th Annual AAAI Conference on Artificial Intelligence* (2024)

[12] A. Hulot, J. Chiquet, F. Jaffrézic, G. Rigaill, Fast tree aggregation for consensus hierarchical clustering, BMC bioinformatics **21**, 120 (2020).

[13] M.R. Smith, Using information theory to detect rogue taxa and improve consensus trees, Systematic Biology **71**, 1088 (2022).

[14] R.N. McArthur, A.N. Zehmakan, M.A. Charleston, Y. Lin, G. Huttley, Spectral cluster supertree: fast and statistically robust merging of rooted phylogenetic trees, Frontiers in Molecular Biosciences **11**, 1432495 (2024).

[15] W.A. Akanni, M. Wilkinson, C.J. Creevey, P.G. Foster, D. Pisani, Implementing and testing bayesian and maximum-likelihood supertree methods in phylogenetics, Royal Society open science **2**, 140436 (2015).

[16] M.D. Karcher, C. Zhang, F.A. Matsen IV, Variational supertrees for bayesian phylogenetics, Bulletin of Mathematical Biology **86**, 114 (2024).

[17] J.R. Ribeiro, A. Ferrari, Phylogenetic analysis of the belostoma plebejum group sensu nieser (insecta, hemiptera, belostomatidae): the effect of adding continuous characters on its accuracy, Arthropod Systematics & Phylogeny **81**, 1 (2023).

[18] T.P. Ruschel, F.M. Bianchi, L.A. Campos, G.S. Carvalho, Total evidence analysis elucidates the tangled systematic scenario within fidicinini (hemiptera: Auchenorrhyncha, cicadidae), Arthropod Systematics & Phylogeny **81**, 35 (2023).

[19] M. Wilkinson, J.A. Cotton, J.L. Thorley, The information content of trees and their matrix representations, Systematic Biology **53**, 989 (2004).

[20] J. Jansson, W.K. Sung, S.A. Tabatabaee, Y. Yang, A Faster Algorithm for Constructing the Frequency Difference Consensus Tree, in *41st International Symposium on Theoretical Aspects of Computer Science (STACS 2024)* (Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024), pp. 43–1

[21] S.R. Alam, M.M. Mahmud, M.S. Rahman, A Heuristic for Maximum Greedy Consensus Tree Problem, in *2022 12th International Conference on Electrical and Computer Engineering (ICECE)* (IEEE, 2022), pp. 128–131

[22] E. Mossel, M. Steel, Majority rule has transition ratio 4 on yule trees under a 2-state symmetric model, Journal of theoretical biology **360**, 315 (2014).

[23] V. Nikkhah, S.M. Babamir, S.S. Arab, Estimating bifurcating consensus phylogenetic trees using evolutionary imperialist competitive algorithm, Current Bioinformatics **14**, 728 (2019).

[24] J.H. Degnan, N.A. Rosenberg, Discordance of species trees with their most likely gene trees, PLoS genetics **2**, e68 (2006).

[25] J.H. Degnan, M. DeGiorgio, D. Bryant, N.A. Rosenberg, Properties of consensus methods for inferring species trees from gene trees, Systematic Biology **58**, 35 (2009).

[26] D.L. Swofford, P.J. Waddell, J.P. Huelsenbeck, P.G. Foster, P.O. Lewis, J.S. Rogers, Bias in phylogenetic estimation and its relevance to the choice between parsimony and likelihood methods, Systematic biology **50**, 525 (2001).

[27] N. Tahiri, B. Fichet, V. Makarenkov, Building alternative consensus trees and supertrees using k-means and robinson and foulds distance, Bioinformatics **38**, 3367 (2022).

[28] P. Gambette, A. Guénoche, Bootstrap clustering for graph partitioning, RAIRO-Operations Research-Recherche Opérationnelle **45**, 339 (2011).

[29] N. Aguse, Y. Qi, M. El-Kebir, Summarizing the solution space in tumor phylogeny inference by multiple consensus trees, Bioinformatics **35**, i408 (2019).

[30] M. Ester, H.P. Kriegel, J. Sander, X. Xu et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in *kdd* (1996), Vol. 96, pp. 226–231

[31] M. Fuchs, W. Höpken, in *Applied Data Science in Tourism: Interdisciplinary Approaches, Methodologies, and Applications* (Springer, 2022), pp. 129–149

[32] D.F. Robinson, L.R. Foulds, Comparison of phylogenetic trees, Mathematical biosciences **53**, 131 (1981).

[33] F. Critchley, B. Fichet, in *Classification and dissimilarity analysis* (Springer, 1994), pp. 5–65

[34] W.H. Day, Optimal algorithms for comparing trees with labeled leaves, Journal of classification **2**, 7 (1985).

[35] H.P. Kriegel, E. Schubert, A. Zimek, The (black) art of runtime evaluation: Are we comparing algorithms or implementations?, Knowledge and Information Systems **52**, 341 (2017).

[36] E. Schubert, J. Sander, M. Ester, H.P. Kriegel, X. Xu, Dbscan revisited, revisited: why and how you should (still) use dbscan, ACM Transactions on Database Systems (TODS) **42**, 1 (2017).

[37] R.K. Thauer, A.K. Kaster, H. Seedorf, W. Buckel, R. Hedderich, Methanogenic archaea: ecologically relevant differences in energy conservation, Nature Reviews Microbiology **6**, 579 (2008).

[38] K. Katoh, J. Rozewicki, K.D. Yamada, Mafft online service: multiple sequence alignment, interactive sequence choice and visualization, Briefings in bioinformatics **20**, 1160 (2019).

[39] A. Criscuolo, S. Gribaldo, Bmge (block mapping and gathering with entropy): a new software for selection of phylogenetic informative regions from multiple sequence alignments, BMC evolutionary biology **10**, 1 (2010).

[40] M. Torres, J.O.d. Silva, Parallel solution based on collective communication operations for phylogenetic bootstrapping in PhyML 3.0, in *Brazilian Symposium on Bioinformatics* (Springer, 2018), pp. 133–145

A Supplementary Material

A.1 Real dataset

The six colors indicate distinct methanogenic orders: *Methanopyrales* (considered an outlier) in black, *Methanobacteriales* in blue, *Methanococcales* in purple, *Methanomicrobiales* in green, *Methanocellales* in red, and *Methanosarcinales* in orange.

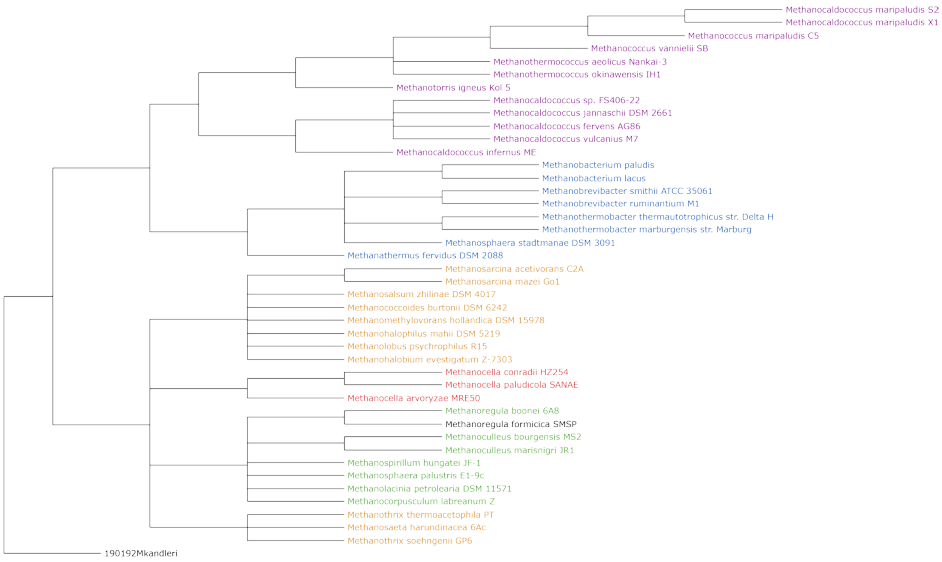

Figure A1. Consensus trees of the evolutionary relationships were constructed based on 35 genes present as single copies across the genomes of 43 analyzed methanogens, comprising a total of 46,657 positions.

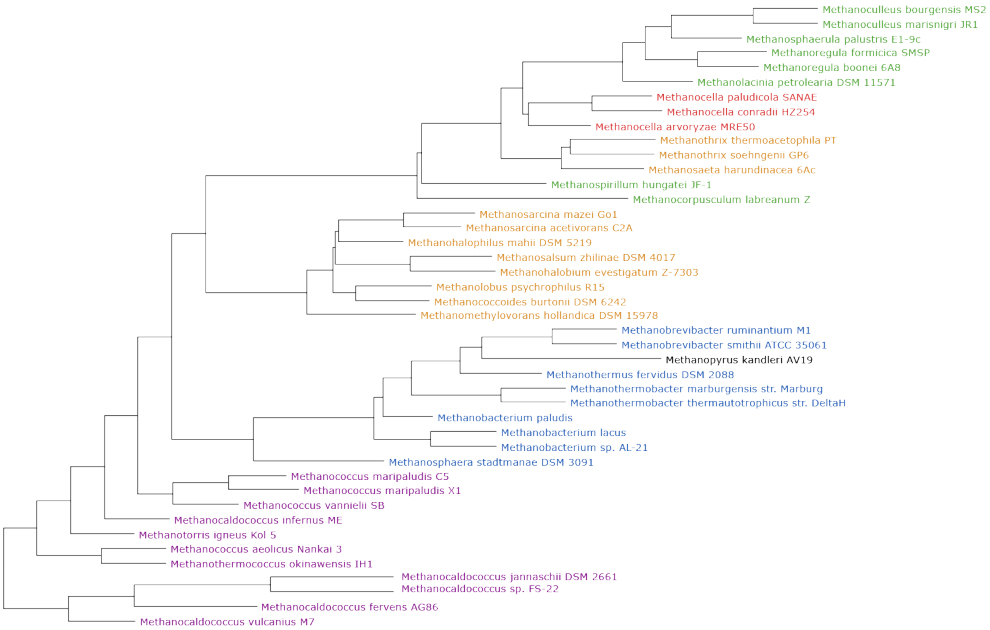

Figure A2. First cluster of alternative consensus trees generated by PhyDBSCAN.

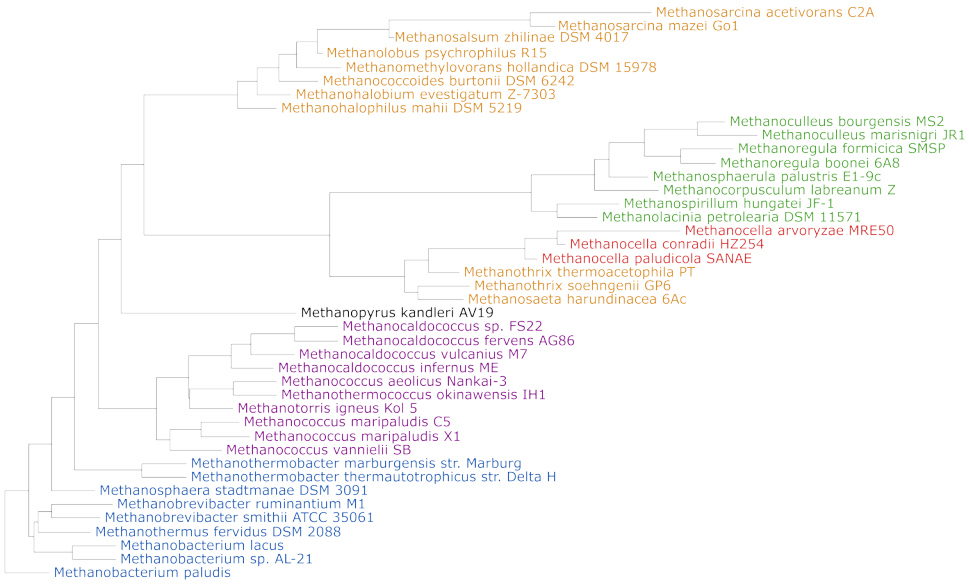

Figure A3. Second cluster of alternative consensus trees generated by PhyDBSCAN.

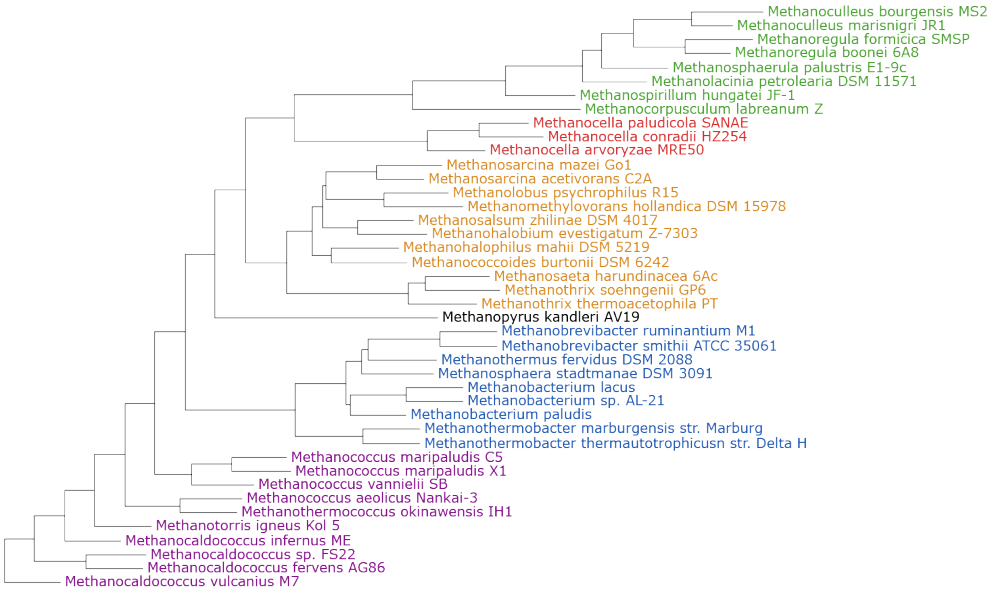

Figure A4. Third cluster of alternative consensus trees generated by PhyDBSCAN.

A.2 Counterexample of correctness

Table A1. Pairwise matrix of Robinson–Foulds (RF) distances between nine rooted trees defined on the same set of twelve leaves.

	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9
T_1	0	0.2	0.2	0.4	0.6	0.6	1	1	1
T_2	0.2	0	0.2	0.5	0.5	0.6	1	1	1
T_3	0.2	0.2	0	0.5	0.6	0.5	1	1	1
T_4	0.4	0.5	0.5	0	0.2	0.2	1	1	1
T_5	0.6	0.5	0.6	0.2	0	0.2	1	1	1
T_6	0.6	0.6	0.5	0.2	0.2	0	1	1	1
T_7	1	1	1	1	1	1	0	0.2	0.2
T_8	1	1	1	1	1	1	0.2	0	0.2
T_9	1	1	1	1	1	1	0.2	0.2	0

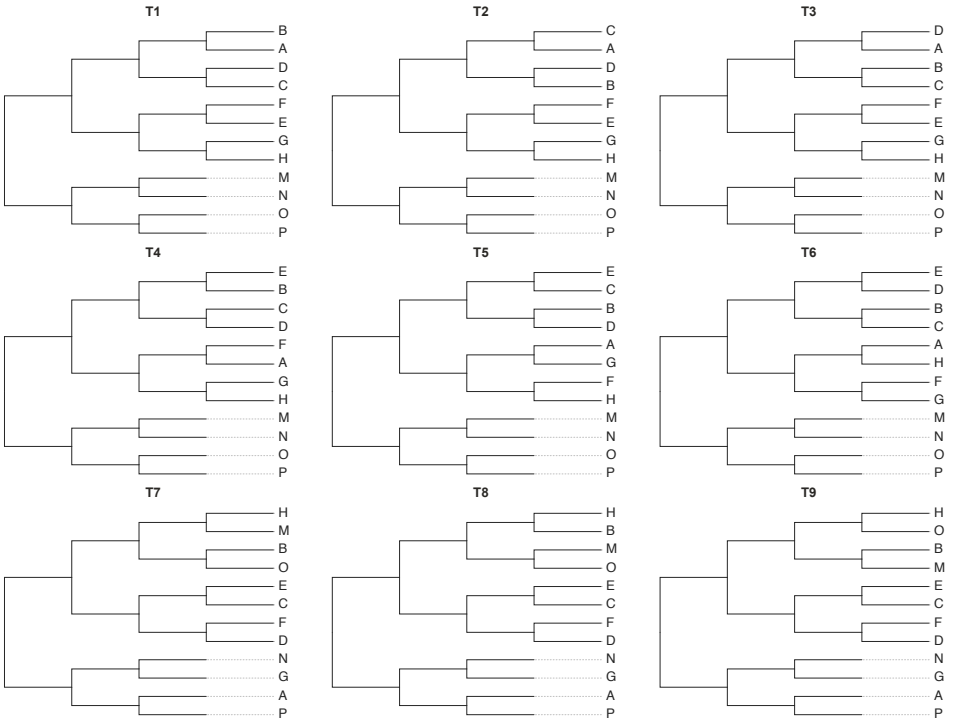

Figure A5. Counterexample illustrating that PhyDBSCAN does not always produce the correct clusters. The example consists of nine rooted trees defined on the same set of twelve leaves.