PhytClust: efficient and optimal node clustering in phylogenetic trees

Katyayni Ganesan[1], Elisa Billard[1], Tom L. Kaufmann[1,2,3], Cody B Strange[1], Maja C. Cwikla[1], Adrian Altenhoff[4], Christophe Dessimoz[4], Roland F. Schwarz[1,2]

[1]Institute for Computational Cancer Biology (ICCB), Center for Integrated Oncology (CIO), Cancer Research Center Cologne Essen (CCCE), Faculty of Medicine and University Hospital Cologne, University of Cologne, Cologne, Germany.
[2]BIFOLD — Berlin Institute for the Foundations of Learning and Data, Berlin, Germany.
[3]Department of Electrical Engineering and Computer Science, Technische Universität Berlin, Berlin, Germany
[4]Department of Computational Biology, University of Lausanne, Lausanne, Switzerland, Swiss Institute of Bioinformatics, Lausanne, Switzerland

Abstract

Phylogenetic trees play a fundamental role in elucidating evolutionary relationships among taxa. Clustering taxa remains a major challenge across diverse biological domains such as cancer genomics, microbial systematics, and phylogenomics. Several methods partition taxa in phylogenetic trees into clusters, but these approaches face key limitations. Many rely on arbitrary distance thresholds that are contingent on a-priori information. These thresholds can be hard to ascertain, are subject to bias, and may vary across studies—hindering meaningful interpretation and comparison. More broadly, many methods lack rigorous definitions of what constitutes a cluster, and depend on heuristics that restrict the cluster search space since enumerating all possible clusters is computationally infeasible for large trees.

Here, we present PhytClust, a threshold-free algorithm that partitions taxa in phylogenetic trees into monophyletic clusters. PhytClust provides an exact solution to the problem of finding an optimal set of clusters in a phylogenetic tree that minimizes the total intra-cluster branch lengths for a given number of clusters. It then determines the optimal number of clusters using a validity index. PhytClust yields an exact and efficient clustering solution that reflects the tree's topology and genetic distances.

In simulated datasets, PhytClust outperforms existing methods in both speed and accuracy and scales to trees with more than a hundred thousand taxa. We apply PhytClust across cancer genomics, avian phylogenomics, bacterial and archaea phylogenetics, and plant genomics to demonstrate PhytClust's varied applicability. By providing a standardized method for taxa clustering within phylogenetic trees, PhytClust yields reproducible, optimal and computationally efficient clusters.

Keywords

Clustering, Phylogenetics, Taxonomic grouping, Subclones

Introduction

Phylogenetic trees are the primary means of quantitatively analyzing relationships among taxa and underpin many areas of biological research[1]. Phylogenetic inference collapses large genetic datasets into a coherent evolutionary narrative[2], enabling the identification of branching patterns and genealogical hierarchies, reconstruction of ancestral states, and tests of evolutionary hypotheses[3,4]. Phylogenies are widely used to track pathogen trajectories, dissect tumour evolution, reveal trait diversification across species and inform bioconservation strategies [5–7].

A recurring challenge in these applications is identifying robust subgroups of leaf nodes within a phylogeny. These subgroups or clusters are typically sets of closely related taxa that are meaningfully distinct from other taxa in the tree. In species evolution, such clusters can naturally emerge as the result of external factors such as changing environmental conditions leading to adaptive radiation or a change in the interplay of anagenesis and cladogenesis, or genetic events such as gene duplications **(Fig. 1a)**. In systematics, clusters are often aligned with taxonomic ranks, but manual curation is difficult for closely related species lacking clear synapomorphies and can yield uneven diversity across equivalent ranks[8]. Additionally, as new species are added to the tree of life, classification becomes more difficult. For example, in the recently discovered *Asgard* archaea, subfamilies were curated manually by the researchers, and these classifications have to be re-evaluated any time new species are added to the tree[9–11]. Similarly, in cancer research, tumour phylogenies are reconstructed based on somatic mutations found in cancer cells and *subclones*, i.e. cancer cells with similar, not necessarily identical, genetic profiles, are often identified manually or using arbitrary thresholds[12–14]. Analogous issues arise when defining *operational taxonomic units* in microbiome research or delineating viral lineages in epidemiology[15].

Systematic, objective, and reproducible clustering directly on tree geometry therefore remains an essential goal. On ultrametric trees, clusters can be defined by a vertical cut at a fixed distance from the root or tip[16]. In non-ultrametric trees however, no simple solution exists and most algorithms depend on manual intervention or user-defined thresholds that can introduce bias and limit reproducibility[16,17].

Existing tools illustrate these trade-offs. Early approaches for transmission clusters, such as ClusterPicker and PhyloPart, define a monophyletic clade as a cluster if within-clade patristic distances fall below a user-specified bound (maximum for ClusterPicker; median for PhyloPart)[18,19]. Later methods, including TreeCluster, enforce bounds such as maximum diameter or total within-cluster branch length on each cluster[16]. PhyCLIP views each internal-node subtree as a candidate cluster and uses size and diversity constraints and statistical tests within an integer-linear program[17]. Across these strategies, user-specified parameters remain central, complicating reproducibility and cross-study comparison. At the same time, thresholds often serve as proxies that cap computational burden in the vast space of possible tree partitions.

A more recent approach, AutoPhy[20], reduces threshold dependence by clustering in an embedded distance space. It projects patristic distances using Uniform Manifold Approximation and Projection[21], fits a Gaussian mixture model[22] with the number of components chosen by the Bayesian information criterion[23], and then splits provisional clusters into monophyletic clades when necessary. While this reduces explicit cutoffs, runtime can vary with topology and tree size (for example, over 100 minutes on a ~3,900-tip SARS-CoV-2 tree)[20]. Collectively, these limitations motivate methods that (i) operate directly on phylogenies, (ii) avoid user-defined thresholds, (iii) are exact for a given model, and (iv) scale to large trees.

To address these limitations, we have developed PhytClust. PhytClust divides any rooted phylogenetic tree into disjoint sets of monophyletic clusters (clades) based on a well-defined mathematical criterion of minimising the total within-cluster branch length. PhytClust implements an exact dynamic programming (DP) that computes the global optimum for any specified number of clusters *K*. PhytClust then selects the best number of clusters (k*) from candidate clusters $k \in [2, N]$, using a tree-specific goodness-of-fit quality index. On simulated data, PhytClust outperforms existing methods while remaining competitive in runtime. We show PhytClust's applicability on real-world trees including single-cell cancer phylogenies, species trees from various biological domains including bacteria, archaea, plants and birds, and on gene trees where it accurately identifies gene duplication events. By anchoring clusters to monophyletic clades and optimizing a transparent, branch-length–based objective, PhytClust yields standardized, reproducible partitions that respect ancestry and minimize within-cluster divergence.

Results

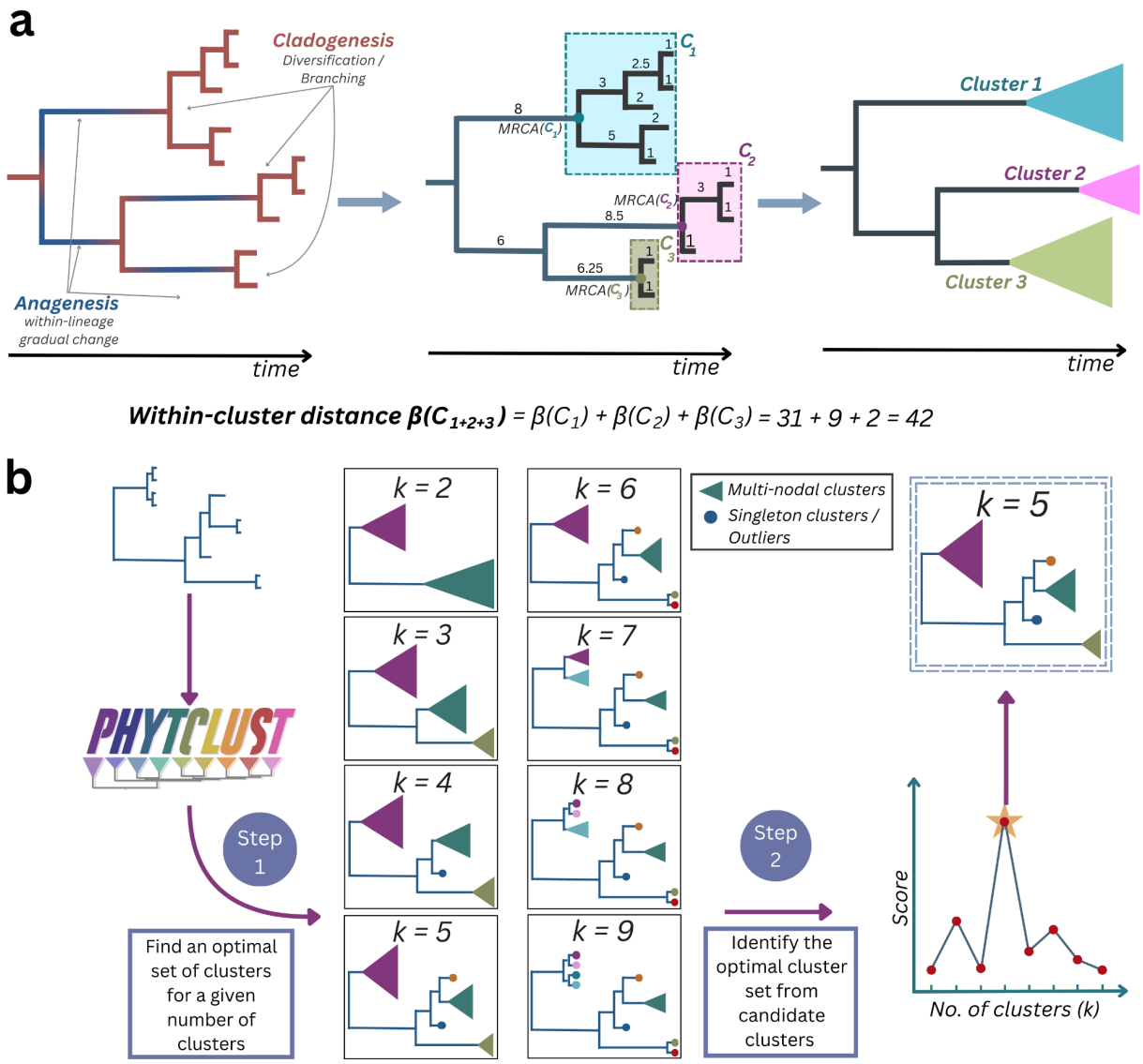


Fig. 1: *(a) An illustrative example of a phylogenetic tree in which both anagenetic and cladogenetic processes are partitioned into three family-level clusters (C1, C2, C3). Clusters are defined by minimizing the cumulative within-cluster distance β, calculated by summing the distances from each taxon to its most recent common ancestor (MRCA) within the cluster. (b) PhytClust workflow: Given a phylogenetic tree, PhytClust first identifies the optimal clustering solution for every possible number of clusters in the range [2, N], where N is the number of leaves in the tree. From this collection of candidate solutions, it then selects the globally optimal set of clusters using our cluster quality index.*

PhytClust partitions phylogenetic trees into optimal sets of monophyletic clades

We formalize clustering on rooted phylogenies as the problem of partitioning the leaves into *k* disjoint monophyletic clusters. Enforcing monophyletic clusters anchors these clusters to a shared ancestry rather than ad hoc distance cutoffs. Similar to clustering concepts in Euclidean spaces, we strive for clusters of taxa that are closely related within the clusters, while remaining relatively distant between the clusters. Thus, for any *k* between 2 and *N* (number of leaves), we define the optimal partition as the one that minimizes the total within-cluster branch length β **(Fig. 1a).** Minimizing total within-cluster branch length simultaneously increases the between-cluster branch lengths, thereby indirectly maximizing separation between clusters **(Supplementary Proof. A)**.

For a given value of *k*, identifying the set of monophyletic clusters that minimize β through exhaustive enumeration is only feasible for small trees **(see Methods; Supplementary Fig. 1a)**. Empirical trees typically fall between these extremes[24]. To efficiently obtain optimal partitions for each possible number of clusters *k*, we make use of the tree's natural hierarchical structure. We recursively divide the tree into successively smaller subtrees and construct the global solution from the solutions to these subproblems using dynamic programming **(Supplementary Fig. 1b)**. Briefly, for each node, we compute the minimum β attainable with every possible number of clusters within that node's subtree. These intermediate results are stored so that each subproblem is solved only once. By combining the stored results from descendant subtrees, we iteratively determine the optimal partitions for each parent node, continuing upward through the tree until the global optimum at the root is obtained.

From the globally optimal partitions for each possible number of clusters *k*, PhytClust identifies the optimal number of clusters (k^*) by using a cluster-quality index combining two known indices—the Calinski–Harabasz (CH) and the elbow index (EL1)—and adapting them for rooted trees **(see Methods, Fig 1b)**[25,26]. The cluster-quality index enables the selection of partitions that balance compactness within clusters and separation between clusters. The total branch-length cost $\beta(k)$ computed by our dynamic programming procedure thereby corresponds to the within-cluster term $W(k)$ of the CH index **(see Methods and Supplementary Proof A)**. This relationship allows CH to be computed directly from $\beta(k)$ without additional traversals over the tree, limiting the computational complexity and allowing for efficient estimation of the optimal number of clusters k^*. The EL1 index further highlights the point of diminishing returns in $\beta(k)$, where additional clusters provide little improvement in within-cluster compactness **(Methods)**.

Despite this, selecting a single global optimum k^* remains a heuristic and depends on the goals of the analysis. Real-world phylogenies often contain meaningful partitions at several hierarchical levels. In biological taxonomy, for example, organisms are organized into families, orders, and classes, each corresponding to a distinct level of evolutionary divergence. Similarly, in viral or microbial phylogenies, one may wish to identify both broad lineages and fine-grained strain clusters. Because a single *k* cannot capture all relevant levels at once, PhytClust extends beyond the global optimum to report partitions across the tree's hierarchy. To do this, we

evaluate our quality index over $k \in [2,N]$ using log-spaced bins **(Methods)** and identify local maxima that represent optimal partitions at each clade level. The resulting family of partitions summarizes structure across the full range of levels in the tree and allows users to interpret clusters according to their specific analytical goals without manually specifying k.

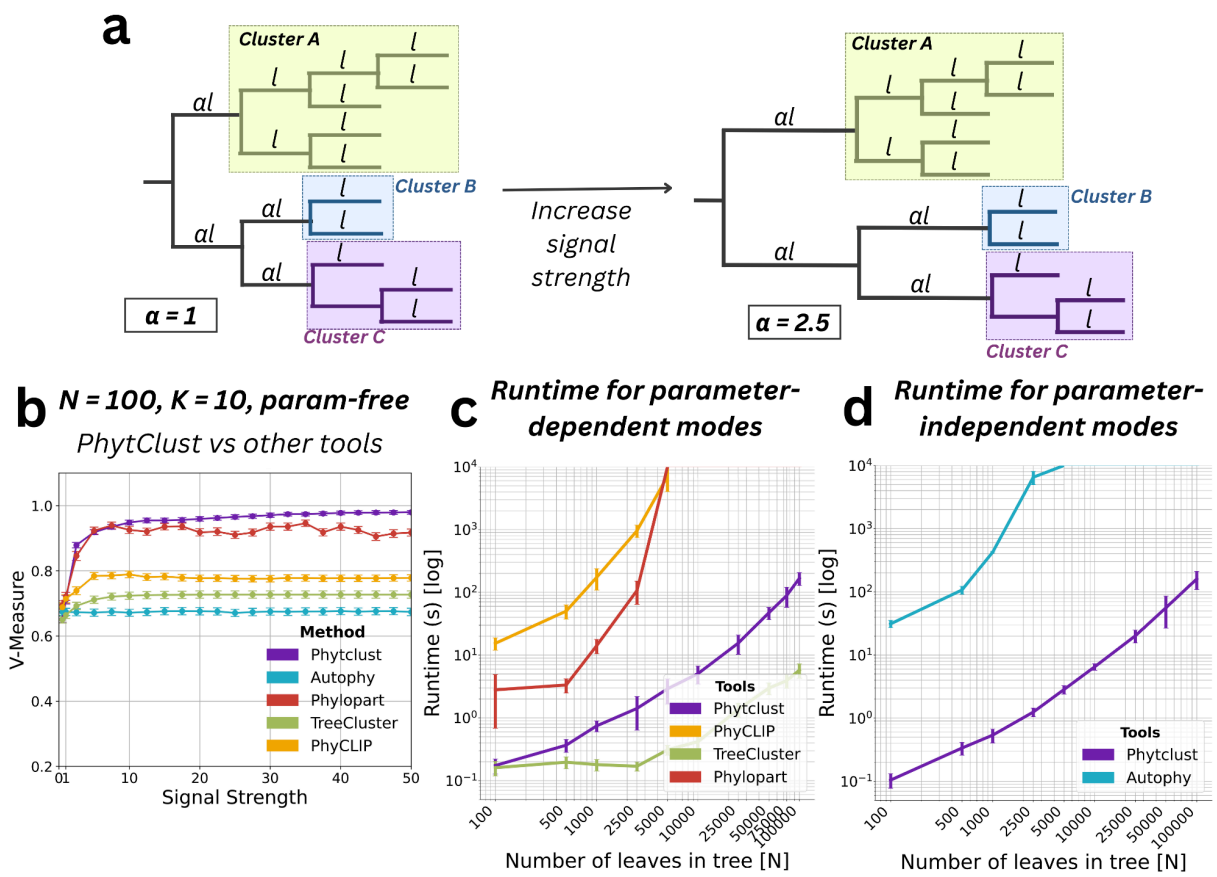


Fig 2: *(a) Simulation setup: Random trees were generated with initial branch length of l = 1 and α = 1, and K random monophyletic target clades were assigned. Signal strength α, defined as the ratio of average between- to within-cluster branch length, was increased on extra-cluster branches. (b) Simulation results reporting V-Measure for PhytClust, AutoPhy, PhyCLIP, TreeCluster and PhyloPart of trees with N =100, K = 10. (c) Runtime comparison of software with a parameter-dependent mode (PhytClust, TreeCluster, PhyloPart, PhyCLIP) (d) Runtime comparison of software with a parameter-independent mode (PhytClust, AutoPhy).*

PhytClust outperforms other software in speed and accuracy on simulated trees

We evaluated PhytClust on simulated trees with tunable signal strength α, defined as the ratio of average between- to within-cluster branch length. For each condition defined by a fixed number of taxa N and true clusters K, we generated 100 rooted binary topologies drawn uniformly from all possible trees[24]. We then selected K disjoint monophyletic target clades, assigned all branches an initial length of one, and increased the between-cluster separation by multiplying branches ancestral to each true clade by $\alpha \in [1,50]$ **(Fig. 2a; Methods)**.

As a null baseline, we generated 100 random monophyletic *K*-partitions with $\alpha = 1$ for each tree. For these random partitions, the V-measure remained close to 0.5 across all values of α, as expected for random partitions. In contrast, the V-Measure of PhytClust increased steadily with signal strength. For $K = 10$ the median V-measure exceeded 0.90 at $\alpha = 15$ and reached V=1.00 for all replicates once $\alpha > 45$. When $\alpha = 1$, within- and between-cluster distances are indistinguishable, so performance approaches a monophyletic-random baseline determined by the tree topology and cluster size distribution **(Supplementary Fig. 1c, 1d).**

When allowing PhytClust to automatically select the number of clusters (k^*) using our cluster-quality index, we compared its performance with PhyCLIP, TreeCluster, PhyloPart, and AutoPhy **(Methods)**. TreeCluster and PhyloPart each require a single user-defined distance threshold that determines how partitions are made, and their performance depends strongly on this choice. To ensure a fair comparison, we performed a grid search over a wide range of thresholds and reported results that maximized the silhouette score[27] **(Fig. 2b; Methods).** PhyCLIP was run across a broad range of its parameters, and its intermediate-resolution mode was used to automatically select one clustering from the multiple solutions it generates.

At low signal strength ($\alpha = 1$), all methods achieved similar V-measure values V around 0.65, reflecting the lack of separation between true clades **(Fig. 2b)**. As signal increased, AutoPhy remained near $V = 0.66$, TreeCluster and PhyCLIP plateaued around $V = 0.70$ and $V = 0.80$ by $\alpha = 10$, and PhyloPart reached about $V = 0.90$ at $\alpha = 7.5$ with higher variability. PhytClust exceeded $V = 0.95$ already at $\alpha = 15$ and approached $V = 0.97$ for stronger signals, outperforming all other tools even for moderate signal strengths. The same trend was observed for $K = 5$ and $K = 20$ **(Supplementary Fig. 1e)**. For $N = 100$ and $K = 10$, qualitative examples show clear under-clustering at low signal, partial recovery at moderate signal, and exact recovery at high signal **(Supplementary Fig. 1g, 2–4)**. Moreover, when we related PhytClust's performance to the balance of the backbone tree (i.e. the tree with clusters removed) and to the evenness of cluster-size distributions using the Gini index[28], we found that higher Gini values required larger α for PhytClust to reliably recover the target clusters **(Supplementary Fig. 1f)**.

To put our simulation results in the context of real-world phylogenies, we estimated the signal strength α for a variety of viral trees. Using the Nextstrain[29] database, we computed α for established clades in the phylogenies of the dengue, monkeypox, and tick-borne encephalitis[30] viruses, which were 42.8, 40.0, and 22.2, respectively. For an independent coronavirus phylogeny[31] delineating seven major groups, we obtained $\alpha = 32.0$, indicating that the signal strengths present in real datasets are well within PhytClust's high-performing α range **(Methods)**. In the coronavirus phylogeny, PhytClust recovered all seven groups as k^* **(Supplementary Fig. 5a)**. For dengue, it recovered the four serotypes[32] **(Supplementary Fig. 5b)**. For tick-borne encephalitis virus, the automatically selected k^* matched the standard clades[30] except for one split and fixing $K = 8$ yielded exact concordance **(Supplementary Fig.**

6a). For monkeypox, k^* identified three of the four named clades[33] and subdivided the largest, while setting $K = 4$ recovered the canonical four **(Supplementary Fig. 6b).**

We next benchmarked runtime across random bifurcating trees containing up to $N = 10^5$ tips (100 replicates per tree size). When using fixed thresholds for PhyloPart, TreeCluster, and PhyCLIP, or a fixed K for PhytClust, TreeCluster was fastest and PhytClust a close second **(Fig. 2c)**. On trees with 10^5 tips, TreeCluster completed in under 10 seconds, while PhytClust required approximately 100 seconds, remaining practical for trees of 10^5 tips. In threshold-free comparisons, we included only PhytClust and AutoPhy, since these methods do not rely on any user-defined cut-offs or parameter grids. Other tools were excluded from this comparison because their runtimes depend strongly on the range of parameters supplied. In this setting, PhytClust ran faster than AutoPhy **(Fig. 2d)**. While the worst-case theoretical time complexity of PhytClust is $O(N^2)$, a power-law fit to measured runtimes across random trees yielded an exponent of ~1.38, indicating a subquadratic scaling **(Supplementary Fig. 10a)**. On Genome Taxonomy Database[34] phylogenies, PhytClust clustered the Archaea (5,869 taxa) in 5 s and the Bacteria (107,235 taxa) in 551 s, demonstrating scalability to genome-scale datasets.

Taken together, these results show that PhytClust combines high accuracy with practical runtime performance, remaining efficient even for trees containing more than 100,000 taxa. We next apply PhytClust to diverse empirical phylogenies to illustrate its scope and utility.

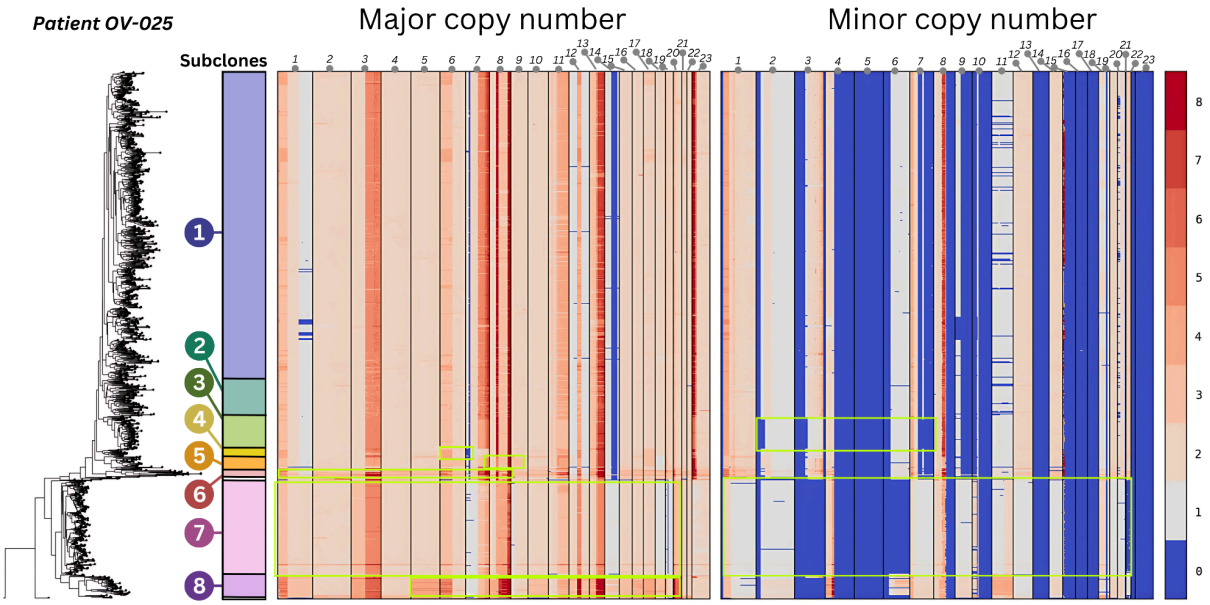


Fig 3: Phylogenetic tree of patient OV-025 showing copy-number profiles and the eight major subclones identified by PhytClust, with distinct copy-number differences highlighted among subclones.

PhytClust identifies meaningful subclones in cancer trees

Cancer cells evolve through point mutations and karyotypic changes such as copy-number (CN) alterations and whole-genome doubling (WGD)[35], generating intra-tumour heterogeneity that complicates treatment. A common challenge in cancer genomics studies is defining subclones in cancer trees reconstructed from single-cell or multi-region bulk sequencing datasets, a task that is still often performed manually[13,14]. Here we instead define subclones objectively using PhytClust.

As an illustrative example, we re-analysed a single-cell phylogeny of a case of high-grade serous ovarian cancer from the SPECTRUM cohort[36]. In patient OV-025, PhytClust inferred eight major subclones and six outlier groups (<12 cells each; ≈0.01% of total) **(Fig. 3).** PhytClust assigned the two key lineages reported by McPherson et al., that had independently acquired two WGD events each, as distinct subclones 6 and 8. The remaining six subclones harbor a single WGD event as previously shown, and exhibit distinct allele-specific CN alterations that clearly separate them from the other subclones. For example, subclone 7 lacks the high-level gains of its sister subclone 8 (two WGDs), instead showing lower copies across 12q/13q and unique LOH on 3q and 6q. Subclone 6 carries arm-level amplifications on 3q and 7q with total copy number of 9 together with a loss-of-heterozygosity (LOH) event on 14q. Even closely related subclones differ in clonality of shared events, for example, on chr1:77–88.5 Mb the allele-specific (3,2) gain occurs in ~90% of subclone 1 cells vs ~56% in subclone 2. Similarly, on chr7:147–149.5 Mb a combined amplification / LOH event with alleles-specific copy number (7,0) occur in ~88% vs ~57% of cells.

We also reanalysed a breast cancer phylogeny (TN-2) originally studied by Minussi et al.[13], with the tree reconstructed using MEDICC2[37]. In this publication, subclonal boundaries were manually chosen to align the tree with sequencing-derived subclones, but several of those groups are not strictly monophyletic, mixing lineages and yielding evolutionary heterogeneity. PhytClust here identified 10 subclones **(Supplementary Fig. 7),** each supported by distinct allele-specific CN features, while preserving evolutionary homogeneity by construction.

Together, these results show that PhytClust yields reproducible, monophyletic subclone calls that respect evolutionary structure and surface biologically interpretable CN events, including independently doubled lineages, offering an objective alternative to manual or threshold-based subclonal delineation in cancer phylogenetics.

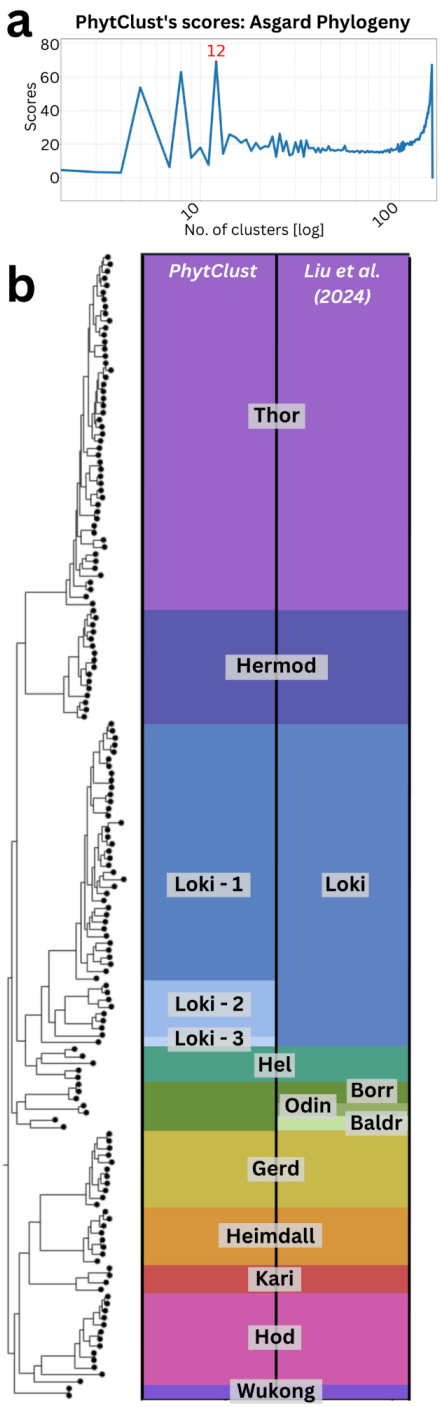


Fig 4: (a) PhytClust scores and optimal solution at k = 12 for the phylogenetic tree of Asgard archaea (b) Phylogenetic tree of the 12 sub-families of the Asgard family introduced by Liu et al. and the subfamilies proposed by PhytClust at k = 12.*

PhytClust outlines hierarchical taxonomic groups in the Asgard phylogeny

The recent discovery of numerous Asgard archaeal lineages has reshaped our understanding of early eukaryotic evolution, but the pace of discovery has also made consistent taxonomic classification increasingly difficult[10,11,38]. As new metagenome-assembled genomes continue to expand this group, reproducible delineation of higher-order clades becomes challenging using traditional, threshold-based approaches[9,39,40]. To assess whether PhytClust can provide a stable, parameter-free alternative, we re-analysed the Asgard phylogeny from Liu et al. (2021b)[9]. Liu et al. assembled 209 single-copy Asgard *clusters of orthologous groups (COGs)*, inferred a maximum-likelihood tree, and delineated phylum-level clades by inspecting bootstrap support and gene-content similarities. Using that same tree as input, PhytClust selected an optimal number of clusters of $k^* = 12$ **(Fig. 4a).**

PhytClust's partitions showed extensive overlap with the classification of Liu et al. (V-measure = 0.95), indicating that tree topology and branch-length information alone largely recapitulate expert curation (**Fig. 4b)**. PhytClust subdivided Lokiarchaeota into three phylum-level clusters, while merging together Borrarchaeota, Baldrarchaeota and Odinarchaeota—consistent with the substantially greater taxonomic diversity of Lokiarchaeota relative to the latter groups combined. Interestingly, the COGs belonging to *Genome Taxonomy Database*-recognized species clusters from Loki-2 all belonged to the family *SOKP01* from the order Signyarchaeales in the Loki lineage[34], lending independent support for a finer partition inside Lokiarchaeota. Without access to the underlying sequence data or domain-specific information, PhytClust reconstructed nearly all expert-defined groups from Liu et al. while clarifying ambiguous boundaries.

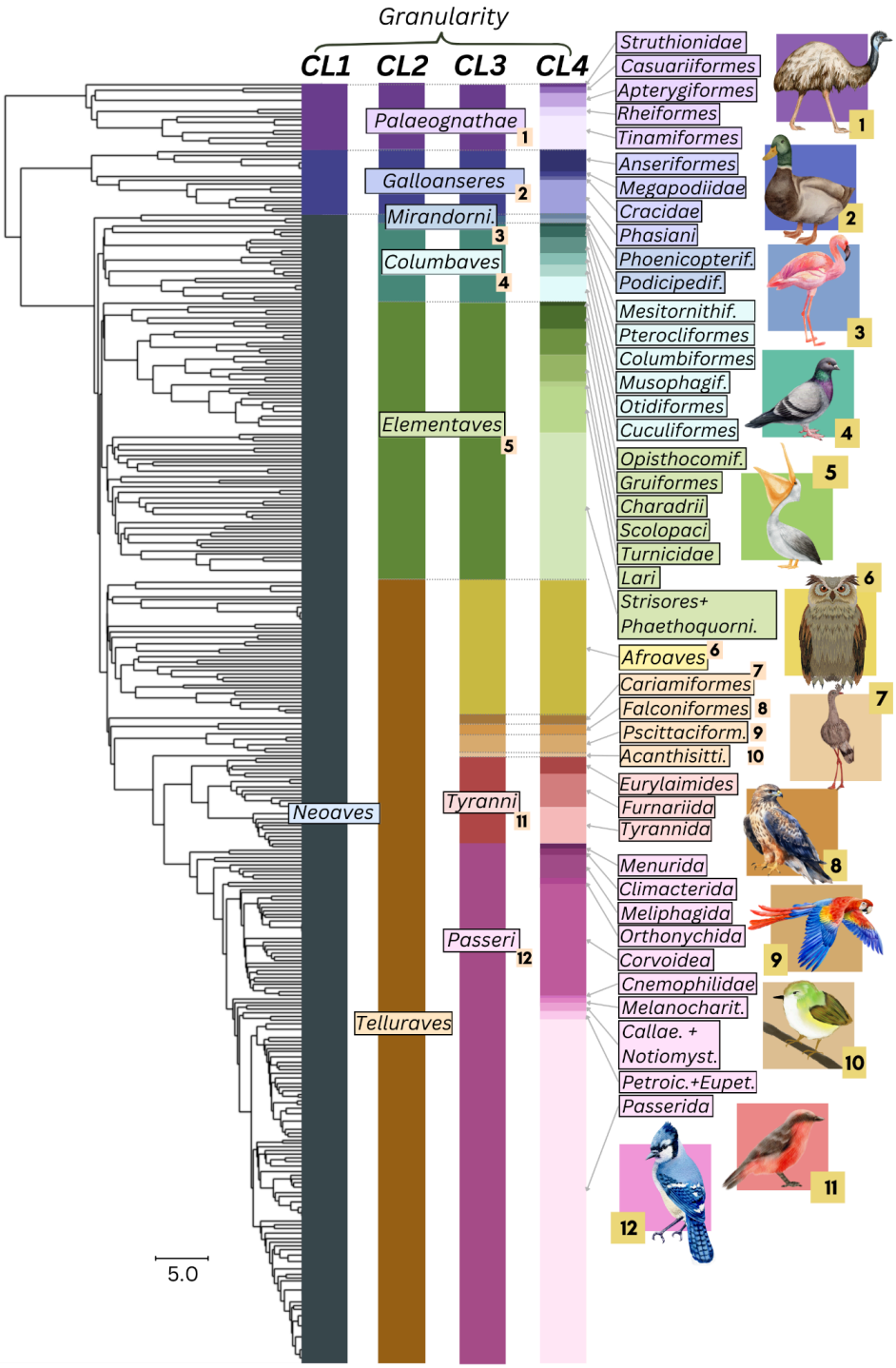


Fig 5: PhytClust was applied to the most comprehensive avian phylogeny available (from Stiller et al.[36]) and its scores were stratified into five clade levels (CLs; four shown). Clusters were labelled according to exact matches with existing taxonomic ranks or, when absent, the closest combination of ranks. Clusters from CL3 are numbered 1–12.

PhytClust identifies common taxonomic groups in Avian phylogeny

Biological classification is inherently hierarchical, with monophyletic groups observable at multiple taxonomic levels—from species and genera to families and orders—each reflecting distinct evolutionary levels. Accurately capturing these nested relationships in large phylogenies remains a key challenge for reproducible taxonomy.

To assess how PhytClust recovers known taxonomic structure across multiple levels of divergence, we applied it to a phylogeny of 363 bird species encompassing 92% of recognized families **(Fig. 5)**[41]. The phylogeny, reconstructed by Stiller et al., was annotated according to established family- and order-level classifications[42,43]. To enable systematic comparison with this reference taxonomy, we stratified the tree into five clustering levels: three corresponding to higher, intermediate, and lower clades; one capturing the classic Palaeognathae–Galloanserae–Neoaves split and a final, fine-grained species-level tier.

At the highest level, PhytClust accurately recovered the three major avian clades—Palaeognathae, Galloanserae, and Neoaves—demonstrating that the method captures broad evolutionary divisions consistent with classical taxonomy. The next level produced five higher-order groups that aligned closely with published classifications, including the proposed Elementaves clade. The third level refined these groups into twelve clusters, differing slightly from the reference classification by splitting Telluraves rather than separating Columbaves and Elementaves. Despite the difference in clusters, overall agreement between PhytClust's clusters and the reference clusters remained strong (V-measure = 0.85) **(Supplementary Fig. 8)**. Moreover, PhytClust achieved a 15% reduction in cumulative within-cluster branch length (reference: 2.3×10^4 ; PhytClust: 1.9×10^4), indicating more internally cohesive clustering.

At the fourth level, PhytClust identified 41 clusters with 10 singletons, compared with 53 clusters and 19 singletons in the reference taxonomy, again producing tighter and more compact groupings (reference: 1.5×10^4; PhytClust: 1.3×10^4). Most clusters aligned closely with existing classifications, and the few deviations were biologically interpretable. For example, PhytClust subdivided the diverse Passeriformes, which is consistent with their known internal heterogeneity, while keeping the smaller Afroaves group intact.

Together, these results indicate that PhytClust robustly recovers hierarchical taxonomic structure across multiple levels of evolutionary divergence, yielding clusters that are both biologically meaningful and quantitatively more cohesive than traditional rank-based classifications. This supports its utility as a consistent, tree-based framework for defining taxonomy without arbitrary rank assignments.

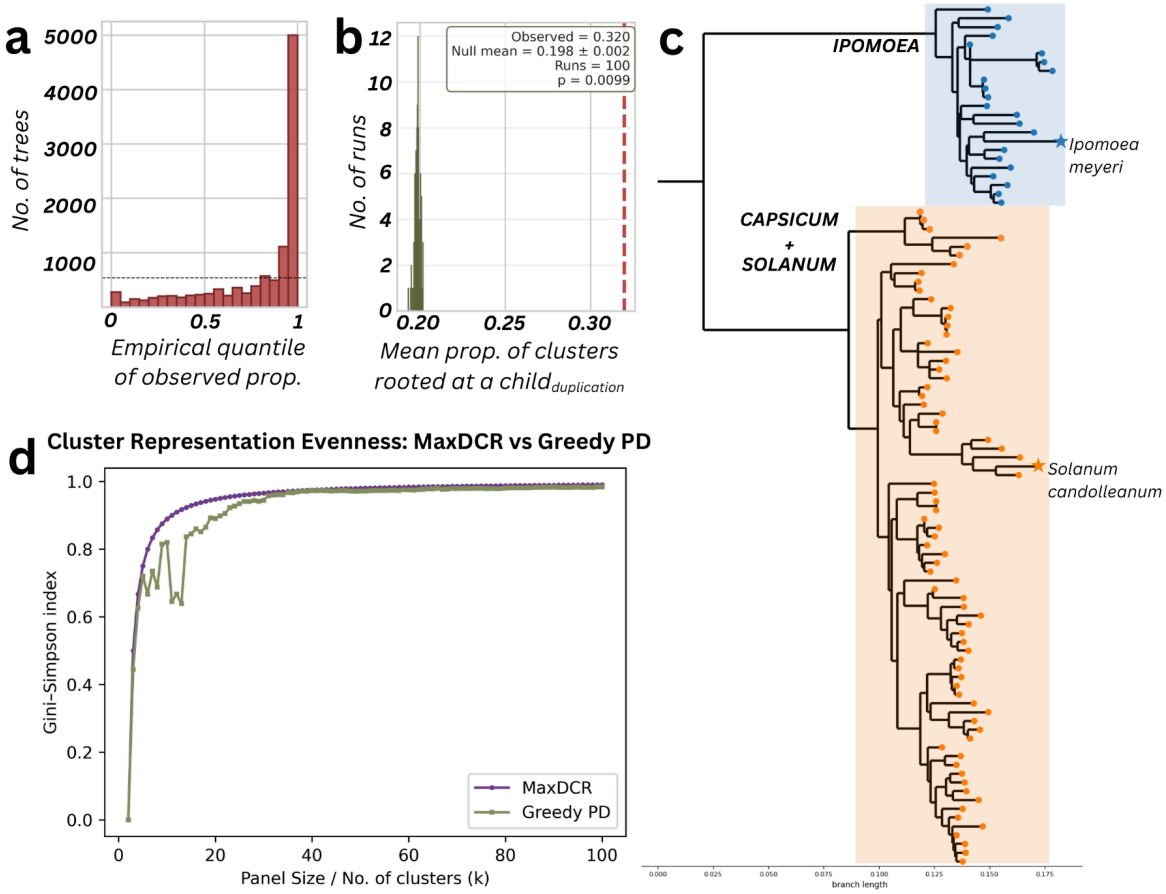


Fig 6: (a) Enrichment of duplication-rooted clusters across trees. Histogram showing, for each tree, the empirical quantile of the observed proportion of PhytClust clusters whose MRCA's parent node is annotated as a duplication. (b) Global enrichment test. Null distribution (100 permutations) of the mean across trees of the proportion of duplication-rooted clusters, with the observed mean indicated by a vertical line. (c) Example section of a phylogenetic tree inferred from 185 species of crop wild relatives of Colombia, highlighting three large clusters (Solanum, Capsicum, Ipomoea) identified by PhytClust. The maximally divergent cluster representatives (MaxDCR) for each cluster are annotated.

Detecting duplication-driven clusters using PhytClust

Gene duplication is a major driver of evolutionary innovation[44]. Following duplication, one gene copy typically retains its ancestral function, while the other experiences relaxed selective constraints and may accumulate mutations[44,45]. This relaxed evolutionary pressure promotes functional divergence and is often reflected in accelerated sequence evolution, producing longer branches for duplicated genes compared to single-copy orthologs[44,45].

These patterns provide a recognizable phylogenetic signal of post-duplication divergence. We hypothesized that such signals could be systematically detected using PhytClust, which clusters trees based solely on branch-length information and topology. Specifically, we tested whether PhytClust preferentially identifies clades whose most recent common ancestor (MRCA) descends from duplication events.

We analysed 15,693 phylogenetic gene trees from the PANTHER database [46], spanning 143 species across the Tree of Life. Each tree was reconciled with the corresponding species phylogeny and annotated with speciation and duplication events. As expected, branches following duplications were significantly longer than those following speciations (mean difference = 0.209, Wilcoxon signed-rank test, $p < 1 \times 10^{-16}$) **(Supplementary Fig. 9a)**, consistent with previous studies[47–49].

We applied PhytClust to the filtered set of 10,660 trees, allowing it to automatically select the optimal number of clusters k^* over the full range of $k \in [2, N]$. PhytClust showed a strong enrichment for duplication-rooted clusters (**Fig. 6a**; Methods). Nearly 48.9% of all trees (5,212) had enrichment scores at or above the 95th percentile of their respective null distributions, and 37.9% (4,040) exceeded the 99th percentile. On average, duplication-rooted clusters comprised 32.0% of PhytClust partitions, compared with 19.8% expected under randomization ($p = 0.010$, one-sided permutation test; **Fig. 6b**). Although only 25.8% of trees (2,751) were individually significant after Benjamini–Hochberg correction[50], reflecting limited per-tree power, the consistent right-shift across trees **(Fig. 6a)** indicates a genome-wide bias toward duplication-rooted clustering.

These findings demonstrate that PhytClust can detect biologically meaningful patterns of accelerated evolution directly from tree structure, without explicit duplication annotations. This capacity highlights its utility for large-scale analyses of gene family evolution, where duplication and diversification are pervasive. In the following section, we illustrate how such cluster outputs can inform downstream functional analyses.

Cluster-based optimisation of phylogenetic representative selection

Phylogenetic indices are widely used to quantify biodiversity and guide conservation or sampling decisions. Among these, phylogenetic diversity (PD)—the total branch length spanned by a selected set of taxa—aims to maximize retained evolutionary history[51–53].
However, PD is agnostic to which taxa contribute to that history. Selecting several closely related species from a deep clade can yield high PD while overlooking evolutionarily distinct lineages. PD-based criteria also underpin practical applications beyond biodiversity, such as reference panel design, where the goal is to minimize the average distance from each taxon to its nearest selected representative (ADCL)[54].

To address this, we implemented a cluster-based selection strategy using PhytClust. The approach first partitions the phylogeny into clusters, ensuring non-redundant coverage by allowing at most one representative per cluster. This enforces balanced representation across major lineages without fixing the panel size *a priori.* Within each cluster, we then select the MaxDCR (maximally divergent cluster representative)—the taxon farthest from the cluster's MRCA—to maximize internal diversity.

Applied to a 185-species phylogeny of Colombian crop-wild relatives spanning 13 families in 11 clades[55], PhytClust yielded 11 clusters **(Supplementary Fig. 9b),** including *Ipomoea, Capsicum*, and *Solanum* **(Fig. 6c)**. Using MaxDCR, the resulting representative sets showed markedly improved evenness of phylogenetic coverage relative to a standard greedy-PD baseline, while maintaining near-identical total PD. Across panel sizes $k \in [2,100]$, MaxDCR achieved higher Gini–Simpson[56] evenness of represented clusters **(Fig. 6d)**, indicating that PD alone tends to overrepresent a few clades. Meanwhile, total PD differed by only −1–2% (median) across k, showing that greater representational balance does not compromise evolutionary breadth.

By integrating PhytClust-derived clusters with established diversity indices, we obtain cluster-aware representative panels that reduce coverage gaps, improve interpretability, and retain maximal phylogenetic diversity.

Discussion

Splitting phylogenetic trees into optimal monophyletic partitions is a common problem in many biological domains. Empirical phylogenies rarely contain clean, perfectly nested groups. Modern trees are often non-ultrametric (rate heterogeneity, heterochronous sampling), while incomplete lineage sorting, hybridization/introgression, horizontal transfer, and recombination can further hinder cluster identification. Despite this, the criteria for what defines an optimal cluster vary between tools and algorithms. PhytClust uses an intuitive cluster definition akin to that found in traditional clustering algorithms on Euclidean spaces: the minimization of within-cluster cophenetic distances.

Several other approaches share conceptual similarities but differ in implementation. AutoPhy[20] and PhyCLIP both aim to group taxa that are close together but AutoPhy does so via an embedding that is dependent on Uniform Manifold Approximation and Projection and subsequent Gaussian Mixture Model settings, while PhyCLIP[17] enforces compactness and separation via user-defined thresholds. In contrast, TreeCluster[16] and PhyloPart[19] define clusters by a single global distance cutoff on the tree, grouping taxa whose pairwise distances stay below that threshold as clusters. A drawback of threshold-based approaches is that trees can live on different branch-length scales (for example, substitutions/site vs. time) and are sensitive to rooting, shifting much of the burden onto users to justify parameter choices. We ran broad silhouette based sweeps for TreeCluster and PhyloPart to avoid cherry picking, but silhouette may not be the best fit for phylogenetic structure and wider sweeps increase runtime, which underscores the value of a method that works threshold-free. While the optimal cluster definition inevitably depends on the application, PhytClust consistently reproduces partitions similar to those obtained by expert manual curation, suggesting that its criterion aligns well with human expectations of biologically meaningful groups. Although our empirical estimates of the signal strength α were obtained from viral phylogenies, the high V-measure between PhytClust's clusters and annotated groups in non-viral datasets from the Avian and Asgard archaea phylogenies suggest that similarly strong signal ranges also arise in non-viral phylogenies.

Despite this, several limitations remain. While the identification of the optimal clustering for a given number of clusters *k* is exact, the identification of the optimal number of clusters is a heuristic and remains subject to human interpretation. Our stratification of the search space of the number of clusters into distinct granularity domains (low, medium and high) alleviates this issue but maintains a certain level of arbitrariness. In practice, in our use cases it was often clear which level to use for which task but this might not always be the case and human expert knowledge is required to make the best use of the tool.

Although TreeCluster completes faster in single-threshold runs, PhytClust performs a more comprehensive optimisation by computing solutions for all possible *k* values. Most of its computational cost arises from constructing the dynamic programming table, which simultaneously computes optimal partitions for all possible values of k. Retrieving the optimal solution or any fixed-k partition adds only minimal overhead from backtracking and evaluation of the cluster-quality index. Thus, even within comparable runtimes, PhytClust performs a substantially more exhaustive optimisation.

Computing the optimal clustering also comes at a computational cost. PhytClust scales quadratically with the number of leaf nodes, and despite our efficient implementation allowing clustering of up to 10^5 nodes, it might not be applicable to metagenomics projects with several million taxa[57]. In the future, we plan to implement core components in an efficient, compiled language such as C, to further improve runtime and memory efficiency. Clustering such a large number of leaf nodes comes with other challenges and further complicates selecting the right number of clusters. Here, collapsing clades at certain levels and performing iterative subclustering might be a valuable strategy also from a biological perspective. Finally, PhytClust was designed to identify optimal monophyletic clades, which, in the presence of horizontal gene transfer or incomplete lineage sorting, might or might not be appropriate. Reticulate processes such as hybridization, introgression, and horizontal gene transfer cannot be fully captured by trees alone and ultimately require network models [58], hence extending PhytClust to phylogenetic networks will be a natural next step.

Despite these drawbacks, our simulations and real-world examples demonstrate both the superior performance on well-defined tasks as well as the real-world interpretability of the suggested tree partitions, largely without relying on user-defined thresholds or arbitrary distance cutoffs. Unlike other clustering approaches, PhytClust results are deterministic and reproducible and broadly applicable to a wide range of biological domains.

Methods

Phylogenetic tree: definitions and notation

Let $T = (V,E)$ be a rooted, acyclic tree with root r. Each edge $e \in E$ has length $w(e) > 0$, which represents the evolutionary distance or time. Let $L(T) \subset V$ be the leaves, with $|L(T)| = N$. For nodes $x,y \in V$, let $d(x,y)$ be the sum of branch lengths along the unique path connecting them.

For any node $v \in V$, let T_v be the subtree rooted at v, with leaf set $L(v) \subseteq L(T)$ and let $ch(v) = \{v_1,\ldots,v_d\}$ denote its children. A set $C \subseteq L(T)$ is monophyletic if $C = L(v)$ for some $v \in V$.

Size of monophyletic search space

Let n(v) be the total number of monophyletic partitions of the leaves L(v). Then

$$n(v) = \prod_{j=1}^{d} n(v_j) + 1, \qquad (1)$$

$$n(leaf) = 1,$$

where the first term in Eq. 1 is the product of the number of partitions each of its children and the second term accounts for the vertex and all of its children being in the same partition together. The total for the whole tree is $n(T) = n(r)$.

Let $a_k(v)$ be the number of monophyletic partitions of $L(v)$ that use exactly k clades. Then

$$a_1(v) = 1, \qquad a_k(v) = \sum_{m_1+\ldots+m_d = k;\, m_j \geq k} \prod_{j=1}^{d} a_{m_j}(v_j) \;\; for\; k \geq 2, \qquad (2)$$

with base cases

$$a_1(leaf) = 1, \qquad a_{k>1}(leaf) = 0\,, \qquad (3)$$

At the root, $a_k(T)$ counts the k-clade partitions; clearly $a_1(T) = a_N(T) = 1$. Note that this recursive formula holds for all trees with arbitrary tree topologies.

To characterize the computational difficulty of exact optimisation, we first quantify the size of the solution space: the number $n(T)$ of distinct ways to partition a tree with N leaves into monophyletic clusters. Building on the coefficient-wise recurrence from (2) and Eq. (1), we obtain closed forms for two external topologies for binary trees.

For fully imbalanced (caterpillar) binary trees, inductively, the root has one leaf child and one caterpillar of N-1 leaves for k >= 2 the only feasible split assigns $(1, k - 1)$ clusters to the two children. Hence

$$a_k(T) = 1 \; for\; all\; k = 1,\ldots, N \qquad n(T) = \sum_{k=1}^{N} a_k(T) = N \qquad (4)$$

For perfectly balanced trees with $N = 2^h$ leaves, $n(T) = f(h) = f(h-1)^2 + 1$, where $f(0) = 1$. With $n(T) = f(h) = f([log_2 N])$, the general bound for any phylogenetic tree T with N leaves is:

$$N \leq n(T) \leq f([log_2 N]) \qquad (5)$$

These counts grow from linear to exponential with N depending on topology: linear for caterpillars, exponential in N for perfectly balanced trees **(Supplementary Fig. 1a)**.

Algorithm overview

Finding the optimal clustering for a given *k* clusters:

Our first objective is to find the optimal *K*-cluster partitions for each value of $k \in [1, N]$.
A valid k-cluster partition $C_k = \{c_1, c_2, \ldots, c_k\}$ must satisfy three conditions: 1) each cluster c_i must be monophyletic, 2) the clusters must form a complete partition of all leaves ($\bigcup_{i=1}^{k} c_{\backslash i} = L(T)$), and 3) the clusters must be mutually disjoint ($c_i \cap c_j = \emptyset$ ($i \neq j$)).

Here, we want to find clusters that minimize the total within-cluster distance β. For each cluster c_i, we measure the sum of distances from each leaf $t \in c_i$ to its MRCA:

$$\beta(c_i) = \sum_{t \in c_i} d(t, MRCA(c_i)). \tag{6}$$

The total cost of the partition C_k is

$$\beta(C_k) = \sum_{i=1}^{k} \beta(c_i) = \sum_{i=1}^{k} \sum_{t \in c_i} d(t, MRCA(c_i)). \tag{7}$$

The optimal *k*-cluster partition C_k^* minimizes this sum:

$$C_k^* = \arg\min_{C_k} \beta(C_k) \quad . \tag{8}$$

To efficiently find the minimum-*β* partition for a given *k*, we employ dynamic programming, utilizing the overlapping subproblems in the tree's hierarchical structure. Let *DP[v,m]* denote the minimum cost of partitioning the leaves *L(v)* in the subtree rooted at *v* into *m* monophyletic clusters. We compute these values by dynamic programming in post-order, so the table entry at the root, *DP[r,k],* yields the minimum cost for partitioning the entire tree into *k* clusters, and back-pointers recover C_k^* . The recurrence is:

$$DP[leaf, 1] = 0$$

$$DP[leaf, m > 1] = +\infty$$

$$DP[v, 1] = \sum_{t \in L(v)} d(t, v)$$

$$DP[v, m > |L(v)|] = +\infty$$

$$DP[v, m] = \min_{m_1 + \ldots + m_d = m;\, m_j \geq 1} \sum_{j=1}^{d} DP[v_j, m_j] \tag{9}$$

where $v_1, \ldots, v_d = ch(v)$. An example of the DP table calculation is shown in **(Supplementary Fig. 1b).** Tree parsing and traversal were implemented using the *Bio.Phylo* module from Biopython[59].Pseudocode for the dynamic programming and backtracking is given in the Supplementary Material.

Selecting the optimal number of clusters:
Because minimizing the cumulative within-cluster distances (β) is equivalent to maximizing between-cluster distances (given the total distance remains constant) (**Supplementary Proof A)**, after obtaining candidate partitions $\{c_1, c_2, c_3, \ldots, c_k\}$ for various k values, we evaluate each partition with a combined validity index that balances between-cluster separation and within-cluster compactness.

Since conventional clustering indices assume Euclidean space, we adapt the Calinski-Harabasz (CH) index to phylogenetic trees in two steps [25]. First, we treat the most-recent common ancestor (MRCA) of each cluster as its centroid and the tree root r as the global centroid. Second, since the square root of patristic branch lengths can embed leaf nodes in Euclidean space [60], we use unsquared branch lengths as squared Euclidean norms, replacing sums of squared Euclidean distances with sums of branch lengths.

We define the within and between terms on the tree as

$$W(k) = \sum_{i=1}^{k} \sum_{t \in c_i} d(t, MRCA(c_i)), \qquad TSS = \sum_{t \in L(T)} d(t, r), \; B(k) = TSS - W(k)$$

Our DP objective equals the within term, i.e. $W(k) = \beta(k)$) **(Supplementary Proof A)**. The tree-CH index is therefore

$$CH(k) = \frac{B(k)/(k-1)}{W(k)/(N-k)} = \frac{\beta(1) - \beta(k)}{\beta(k)} - \frac{L(T) - k}{k} \qquad (10)$$

where $\beta(1) - \beta(k)$ represents the between cluster distances weighted by the number of terminal nodes at each internal node, and, the second term regularizes to reduce overclustering by penalizing partitions with minimal gains in separation.

However, CH scores can increase as k grows if within-cluster terms decrease faster than between-cluster terms, pushing the optimum toward larger k values, which might not be a desired resolution for interpretable clusters. To balance this, we multiply CH with the Elbow index (EL1) [26], similarly expressed directly in β:

$$EL1(k) = \frac{\beta(k-1) - \beta(k)}{\beta(k) - \beta(k+1)} \qquad (11)$$

We rank k by $S(k) = CH(k) * EL1(k)$ and choose peaks of $S(k)$ over k ∈ [2, N]. For each peak, we consider both $S(k)$ and its prominence. Prominence is defined as the difference between the peak value and the value of its nearest lower local minimum. Then, we pick

$$k^* = arg\,max_k\,[\,\lambda\,\widetilde{S}(k) \;\; + \;\; (1 \; - \; \lambda)\;\widetilde{P}(k)\;\,] \qquad (12)$$

where $\widetilde{S}(k)$ and $\widetilde{P}(k)$ are the normalized score and prominence, respectively and $\lambda \in [0, 1]$ controls the trade-off between the two measures. We set $\lambda = 0.7$ based on a stability sweep over simulated trees **(Supplementary Fig. 10c)**, selecting the value that yielded the best recovery of the target clustering at the lowest *α*.

Multiple clade levels:

It is important to note that all peaks in *S(k)* are possible solutions. A tree can have multiple meaningful sets of clusters that represent different taxonomic levels, similar to how we hierarchically categorize organisms into phyla, families, orders, etc. Instead of requiring one optimal solution, users may want a fixed number of clade levels (CL) and corresponding clustering solutions at each level.

For this, we provide a granularity-based stratification of *S(k)* values, which in combination with their ranking, provide the optimal partition at each clade level. Dividing the range $k = [2, N]$ into equally-sized bins may be misleading in larger trees, as most meaningful clustering occurs at smaller *k* values. Instead we use log-spaced k by setting bin boundaries $k_i = \lceil N^{i/b} \rceil$ for i = 0,...,b and search k^* within each $\lceil k_{i-1}, k_i \rceil$. For $N = 1000$ leaves and $b = 5$, this yields $k = [2, 4, 16, 63, 251, 1000]$.

Using bootstrap values:

PhytClust can optionally incorporate branch support values (for example, bootstrap proportions or analogous support measures) when evaluating candidate clusters **(Supplementary Fig. 10b)**. When this option is enabled, low-support internal branches contribute an additional penalty to the within-cluster cost. Concretely, we interpret the support at each internal node as a value $s \in (0, 1]$ and augment the corresponding branch length by a term proportional to $-\log s$, scaled by a user-controllable weight. Branches with high support ($s \approx 1$) therefore behave almost like their original lengths, whereas poorly supported branches become effectively longer and are less likely to be grouped within a single cluster. In addition, when the algorithm considers forming a cluster whose root lies at an internal node, the total cost of that cluster is further scaled by a factor that increases as the node's support decreases. Together, these adjustments discourage clusters that rely heavily on weakly supported splits, while leaving strongly supported parts of the tree essentially unchanged. When node support values are missing, we treat them as 100% (i.e. no penalty), so branch support only influences clustering when explicitly provided in the input tree.

Handling polytomies:

When an internal node has >2 children nodes, we resolve it by inserting zero-length edges . Since the added edges have length 0, patristic distances and therefore β are invariant.

Identifying Representative Species:
Phylogenetic Diversity is defined as the sum of edge lengths for a set of taxa in a phylogeny and is used as a diversity measure in ecological systems. Given a PhytClust partition $C = \{c_1, \ldots, c_k\}$, we select one representative per cluster by maximizing the MRCA-to-tip distance, r_i = arg max $_{t \in ci}$ $d(t$, MRCA(c_i)), i.e. the Maximally Divergent Cluster Representative (MaxDCR).

Evaluating performance using synthetic trees

Simulation Setup

We generated random rooted binary trees with N leaves using ETE3[61], under the Proportional to Distinguishable Arrangements[24] (uniform over labeled topologies) distribution. All edges were initialized to length 1. For a chosen *K,* we sampled *K* disjoint monophyletic target clades by recursively splitting on the phylogenetic tree.

We then set signal strength α as the ratio of average extra-cluster to intra-cluster branch length. We multiplied all extra-cluster edges by $\alpha \in [1, 50]$. We used fixed random seeds per replicate for reproducibility.

As a performance metric, we use the V-Measure[62], defined as the harmonic mean of homogeneity and completeness given by

$$V = \frac{2(H \times C)}{H + C}$$

$$Homogeneity\ (H) = 1 - \frac{H(True|Pred)}{H(True)}$$
$$Completeness\ (C) = 1 - \frac{H(Pred|True)}{H(Pred)}$$

Here, we weigh both homogeneity and completeness equally.

A clustering partition satisfies homogeneity if each cluster contains leaf nodes from a single ground-truth class, while completeness is satisfied if all leaf nodes of a given ground-truth class are successfully assigned to the same cluster. We chose the V-Measure over the popular Adjusted Rand Index[63] (ARI), due to ARI's insensitivity towards singleton clusters. [62,64] **(Supplementary Fig. 10e).** Apart from V-Measure, we also provide validation of the simulations using Normalized Mutual Information[65] (NMI) in (**Supplementary Fig. 10f).**

Performance against random clustering

For each $k \in \{5,10,20\}$ we generated 100 PDA trees with N = 100. As a monophyletic-random baseline, we generated 100 random *k*-partitions per tree by sampling *k* disjoint internal nodes (as above) without using branch lengths. PhytClust was run with a fixed number of clusters *k*. We ran the simulations and calculated the V-Measure for each iteration. To provide a worst-case reference, we used the same dynamic programming framework but reversed the optimization

objective to maximize *β(k)* instead of minimizing it. This yields monophyletic partitions that are intentionally poorly separated with respect to our objective. We refer to this baseline as the *worst-separation* (PhytWorst) partition. Together, the random and worst-separation baselines bracket PhytClust's performance between unstructured random clustering and deliberately poor clustering **(Supplementary Fig. 1c)**.

Benchmarking against other phylogenetic clustering tools

We benchmarked PhytClust against several existing phylogeny-based clustering tools on the same simulated trees described above. We considered AutoPhy[22], TreeCluster[16], PhyloPart[32], and PhyCLIP[17]. We excluded ClusterPicker, as prior work has shown that TreeCluster and PhyCLIP generally outperform it[16,17,19]. AutoPhy is parameter-free and takes only a phylogenetic tree as input, so we ran it with default settings.

PhyloPart requires a user-defined percentile threshold on global pairwise patristic distances. Following the strategy used in earlier comparisons with PhyCLIP[18], we ran PhyloPart over a grid of patristic distance thresholds from 0.05 to 0.50 (step 0.05). For each threshold we obtained a clustering and computed the silhouette index on patristic distances; the partition with the highest silhouette score was retained for comparison.

TreeCluster offers a threshold-free mode that maximises the number of non-singleton clusters but, on our simulated trees, this mode tends to produce only clusters of size two. To obtain more interpretable partitions, we instead used the "sum_branch_clade" objective and scanned a grid of thresholds from 1 to 10 (step 1). As for PhyloPart, we computed silhouette scores for each threshold and selected, for each tree, the partition with the maximum silhouette.

For PhyCLIP, we explored a broad parameter range: minimum cluster size fixed at 2, $\gamma \in [0.10, 0.36]$ in steps of 0.05, and FDR $\in \{1, 2, 3, 4, 5\}$. For each parameter combination PhyCLIP produces one or more candidate clusterings; we used its "intermediate resolution" selection mode to choose a single clustering per parameter setting, and then selected the best-performing solution per tree using the silhouette index on patristic distances. For ease of comparison across methods, we treated each singleton leaf as its own cluster when computing external validation metrics.

For all tools, we evaluated clustering performance on each simulated tree using the V-measure (as in the main simulations), as well as the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) **(Supplementary Fig. 10f)**.

To relate PhytClust's performance to underlying tree shape, we computed two additional summary statistics for each simulated tree. First, we defined the "backbone tree" by collapsing each true target cluster to a single tip at its MRCA, thereby removing all branches below the cluster roots while preserving the higher-level topology. We quantified the balance of this backbone tree using the normalized Sackin index[66], computed as the sum of root-to-tip depths of all leaves.

Second, we quantified heterogeneity in true cluster sizes using the Gini coefficient of the cluster-size distribution. Here, the input values were the numbers of leaves in each ground-truth cluster, and the Gini coefficient ranged from 0 (all clusters equal in size) to 1 (all leaves in a single cluster). In **Supplementary Fig. 1f** we relate the V-measure achieved by PhytClust to both backbone imbalance and cluster-size Gini.

Running PhytClust on Real Trees

Signal strengths in real trees

To investigate whether signal strengths where PhytClust's partitioning approaches the ground truth clusters in the simulations are also found in real trees, we ran PhytClust on trees from Nextstrain[29], including the Dengue, Monkeypox and Tick-borne encephalitis virus phylogenies [29,30,33]. We also used a Coronavirus phylogeny from Dong et al.[31]. For the Dengue and Coronavirus phylogenies, PhytClust was run without a fixed *K*. For the monkeypox and tick-borne encephalitis trees, we additionally ran PhytClust with *K* constrained to equal the number of named clades in the respective studies, so that the inferred clusters could be compared directly with the curated nomenclature.

To estimate empirical signal strength for each tree, we used the named clades from Nextstrain (or from Dong et al. for coronaviruses) as reference clusters. For each such clade we identified its MRCA in the phylogeny and defined the empirical signal strength α for a tree as the ratio of average extra-cluster to average within-cluster branch length.

Cancer Phylogenies

The phylogenetic tree, copy number profile and WGD calls for patient OV-025 were obtained from McPherson et al.[36]. In the study, copy number profiles were derived from single-cell whole genome sequencing of ovarian cancer patients. The copy number profiles were further refined and used to reconstruct the patient OV-025's cancer phylogeny using MEDICC2 [37]. Cluster whole-genome-doubling (WGD) purity was defined as the proportion of cells within each subclone that exhibited a WGD state (for example, 1xWGD and 2xWGD). For each cluster, segments present in ≥85% of member cells were retained as frequent events.To identify unique versus shared events across clusters, we built a segment–cluster presence matrix by grouping identical genomic intervals across all clusters. Segments appearing in exactly one cluster were labeled unique, while those shared by ≥2 clusters were labeled shared. Adjacent segments with identical copy states were merged if their intervals were contiguous, producing a minimal non-redundant set of events per cluster. Each merged segment was then classified by copy-number pattern into standard categories (for example, gain, loss, copy-neutral LOH, homozygous deletion, amplification) based on total and allele-specific copy number. For functional annotation, merged segments were intersected with curated oncogene and tumour-suppressor gene (TSG) coordinates using pybedtools. Genes overlapping a unique segment were reported as potential cluster-specific driver alterations. For our second cancer phylogeny example, we obtained the breast-cancer Minussi et al.[13] phylogeny and cluster annotations from Kaufmann et al[37].

Avian Phylogeny

The Avian tree and corresponding family-level annotations were obtained from Stiller et al.[41]. In the study, whole-genome alignments of 363 bird species representing 218 families were used to extract intergenic regions. These intergenic regions were divided into 2-kb windows and the most complete 1-kb segment was selected from each window. For each locus a maximum-likelihood gene tree was estimated and then used to infer a coalescent-based species tree, followed by post-processing. We ran PhytClust on this tree using our clade-level procedure with $b = 5$ log-spaced bins over the range of k, yielding five clade levels (CL1–CL5) from coarse to fine resolution. For each level we used the locally optimal partition identified by our combined cluster validity index.

To compare PhytClust clusters with the taxonomy of Stiller et al., we treated their higher-level groups as reference labels and compared it with PhytClust's CL3 clusters using V-measure, using the same implementation as in the simulation study. To compare within-cluster branch-length costs, we also computed β for the reference taxonomy by treating each taxonomic group as a cluster, identifying its MRCA in the tree, and summing the branch lengths from each member species to that MRCA, analogous to the quantity minimized by PhytClust.

Asgard Phylogeny

The Asgard phylogeny was obtained from Liu et al.[9]. In the study, the tree was inferred by selecting 209 core clusters of orthologous genes (COGs) conserved across the Asgard archaea. Protein sequences corresponding to these COGs were extracted from 162 Asgard genomes, aligned using MUSCLE and the concatenated alignment was then used to infer a maximum-likelihood phylogenetic tree using IG-Tree with the LG + F + R10 model. We ran PhytClust on this tree and allowed it to infer the optimal number of clusters k^*. To quantify agreement with the phylum-level groupings defined by Liu et al., we treated their taxonomic assignments as reference labels and computed the V-measure between these labels and the PhytClust clusters.

Crop Wild Relative phylogeny

The example phylogeny used for finding representative species per cluster was obtained from González-Orozco et al.[55] who assembled a maximum-likelihood tree of 185 Colombian crop wild relatives spanning 13 families and 11 major clades using seven genetic markers and RAxML[67]. We applied PhytClust to this tree and used the global clustering solutions for all k from 2 to 100. For each k we treated the k PhytClust clusters as the units to be covered and selected one representative taxon per cluster using the MaxDCR feature within PhytClust.

As a cluster-agnostic baseline, we implemented a standard greedy phylogenetic diversity (PD) algorithm. For a given panel size m, we first selected the tip with the longest root-to-tip path, and then iteratively added the tip that maximised the marginal gain in PD. PD was measured as Faith's phylogenetic diversity[53], i.e. the sum of branch lengths in the minimal subtree connecting the selected taxa. For each k we computed total PD for the MaxDCR and greedy PD panels and

recorded the relative difference (MaxDCR − greedy) divided by the greedy PD. We summarised these relative differences across *k* using the median.

To quantify how evenly each panel covered the PhytClust clusters, we computed a Gini–Simpson evenness index for each k and method. For a given panel, we counted how many selected taxa fell into each cluster, converted these counts into frequencies across clusters, and calculated the Gini–Simpson index $1 - \sum p_i^2$, where p_i is the fraction of selected taxa drawn from cluster *i*. This index is close to zero when most representatives come from a single cluster and increases as representation becomes more balanced across clusters. We compared Gini–Simpson values for MaxDCR versus the greedy PD baseline across panel sizes $k \in [2, 100]$.

Duplication nodes

We downloaded 15,693 reconciled gene trees from the PANTHER database v18.0 [46], in which each internal node is annotated as either a speciation or duplication event. To compare branch lengths following duplication and speciation events, we summarised branch lengths per tree. For each gene tree we computed the mean length of branches descending from duplication nodes and the mean length of branches descending from speciation nodes. We then used a one-sided Wilcoxon signed-rank test across trees to test whether post-duplication branches are longer than post-speciation branches, treating each tree as a single paired observation.

For each tree we first computed (i) the proportion of duplication nodes among internal nodes and (ii) the ratio of total internal to total terminal branch length. We retained only trees that (i) had at least 10% of internal nodes annotated as duplications, and (ii) had an internal-to-terminal branch-length ratio of at least 0.1, yielding 10,660 trees for downstream analyses. For each retained tree, we ran PhytClust and used the top-ranked global solution (*k**) as the clustering for that tree. For every cluster, we identified its most recent common ancestor (MRCA) and examined the parent node of that MRCA. If this parent was annotated as a duplication event (*"D=Y"* in the PANTHER annotation), we classified the cluster as duplication-rooted. For each tree we then calculated the proportion of clusters that were duplication-rooted.

To test whether duplication-rooted clusters are more frequent than expected by chance, we constructed a tree-specific permutation null model. For each tree we removed all duplication labels from internal nodes, randomly reassigned the *duplication* label to the same number of internal nodes as in the original tree, reran PhytClust, and recomputed the proportion of duplication-rooted clusters. We repeated this random relabelling 100 times per tree. The tree-wise permutation p-value was defined as the fraction of permuted proportions that were at least as large as the observed proportion (with a +1 correction in numerator and denominator to avoid zero p-values). We corrected these p-values across trees using the Benjamini–Hochberg procedure and reported both raw and FDR-adjusted values. For visualization, we also recorded,

for each tree, the empirical quantile of the observed proportion within its 100 permuted values. Under the null these quantiles are approximately uniformly distributed.

To summarise enrichment across trees, we also performed a global permutation test. For each random relabelling we averaged the duplication-rooted proportion across all trees and compared the observed mean to the distribution of these permuted means, again using a right-tailed permutation p-value.

Computational Performance and Technical Framework

Runtime was tested using randomly generated trees with up to 100,000 leaf nodes using the ETE3 library[61]. Runtime was calculated for the entire workflow described in the benchmarking section, i.e. starting with a newick file to processing and saving the results for the optimal clustering solution. Implementation of each software followed the setup for the simulations. Scaling exponents were estimated by linear regression of $\log_{10}T$ against $\log_{10}N$ on the largest trees ($N \geq 5{,}000$). We used the phylogenetic trees for Archaea and Bacteria (release 220) from the GTDB database to test PhytClust's runtime on empirical phylogenies[34].

All benchmarks were run locally on a Lenovo ThinkPad X1 (32GB) with 12th Gen Intel(R) Core(TM) i7-1260P to reflect typical use, but access to a high-performance cluster and additional memory enables substantially larger trees to be processed as well.

Data and Code availability

The source code used to perform all simulations and generate the figures presented in this manuscript is publicly available at **10.5281/zenodo.17903047.** The PhytClust software package is available through Conda or from the official Bitbucket repository at https://bitbucket.org/schwarzlab/phytclust.

Acknowledgements

This work was funded by BMBF as "SATURN3: **S**patial **A**nd **T**emporal Resol**u**tion of Intratumoural Heterogeneity in **3** hard-to-treat Ca**N**cers", Project number 01KD2206L. Roland F. Schwarz is a Professor at the Cancer Research Center Cologne Essen (CCCE) funded by the Ministry of Culture and Science of the State of North Rhine-Westphalia. This work was partially funded by the German Ministry for Education and Research as BIFOLD - Berlin Institute for the Foundations of Learning and Data (ref. 01IS18025A and ref 01IS18037A). This work was supported by the Bruno and Helene Jöster Foundation as "CLONETRAC - Tracking the clonal dynamics of cancer through treatment at the single-cell level". The authors acknowledge support from the Helmholtz Association. We would also like to thank the authors that contributed data to Nextstrain to allow us to use well-reconstructed phylogenies.

References

1. Wiley, E. O. Why Trees Are Important. *Evol. Educ. Outreach* **3**, 499–505 (2010).
2. Kapli, P., Yang, Z. & Telford, M. J. Phylogenetic tree building in the genomic age. *Nat. Rev. Genet.* **21**, 428–444 (2020).
3. Soltis, D. E. & Soltis, P. S. The Role of Phylogenetics in Comparative Genetics. *Plant Physiol.* **132**, 1790–1800 (2003).
4. Scossa, F. & Fernie, A. R. Ancestral sequence reconstruction - An underused approach to understand the evolution of gene function in plants? *Comput. Struct. Biotechnol. J.* **19**, 1579–1594 (2021).
5. Ng, J. & Smith, S. D. How traits shape trees: new approaches for detecting character state‐dependent lineage diversification. *J. Evol. Biol.* **27**, 2035–2045 (2014).
6. Beerenwinkel, N., Schwarz, R. F., Gerstung, M. & Markowetz, F. Cancer Evolution: Mathematical Models and Computational Inference. *Syst. Biol.* **64**, e1–e25 (2015).
7. Li, T. *et al.* Phylogenetic supertree reveals detailed evolution of SARS-CoV-2. *Sci. Rep.* **10**, 22366 (2020).
8. Purvis, A. & Agapow, P.-M. Phylogeny Imbalance: Taxonomic Level Matters. *Syst. Biol.* **51**, 844–854 (2002).
9. Liu, Y. *et al.* Expanded diversity of Asgard archaea and their relationships with eukaryotes. *Nature* **593**, 553–557 (2021).
10. Sun, J. *et al.* Recoding of stop codons expands the metabolic potential of two novel Asgardarchaeota lineages. *ISME Commun.* **1**, 30 (2021).
11. Xie, R. *et al.* Expanding Asgard members in the domain of Archaea sheds new light on the origin of eukaryotes. *Sci. China Life Sci.* **65**, 818–829 (2022).
12. Zeng, L., Warren, J. L. & Zhao, H. Phylogeny-based tumor subclone identification using a

Bayesian feature allocation model. *Ann. Appl. Stat.* **13**, 1212–1241 (2019).

13. Minussi, D. C. *et al.* Breast tumours maintain a reservoir of subclonal diversity during expansion. *Nature* **592**, 302–308 (2021).
14. Tarabichi, M. *et al.* A pan-cancer landscape of somatic mutations in non-unique regions of the human genome. *Nat. Biotechnol.* **39**, 1589–1596 (2021).
15. Schloss, P. D. & Westcott, S. L. Assessing and improving methods used in operational taxonomic unit-based approaches for 16S rRNA gene sequence analysis. *Appl. Environ. Microbiol.* **77**, 3219–3226 (2011).
16. Balaban, M., Moshiri, N., Mai, U., Jia, X. & Mirarab, S. TreeCluster: Clustering biological sequences using phylogenetic trees. *PloS One* **14**, e0221068 (2019).
17. Han, A. X., Parker, E., Scholer, F., Maurer-Stroh, S. & Russell, C. A. Phylogenetic Clustering by Linear Integer Programming (PhyCLIP). *Mol. Biol. Evol.* **36**, 1580–1595 (2019).
18. Prosperi, M. C. F. *et al.* A novel methodology for large-scale phylogeny partition. *Nat. Commun.* **2**, 321 (2011).
19. Ragonnet-Cronin, M. *et al.* Automated analysis of phylogenetic clusters. *BMC Bioinformatics* **14**, 317 (2013).
20. Ortiz-Velez, A. N., Sukumaran, J., Rouzbehani, R. & Kelley, S. T. AutoPhy: Automated phylogenetic identification of novel protein subfamilies. *PLOS ONE* **19**, e0291801 (2024).
21. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **3**, 861 (2018).
22. Reynolds, D. Gaussian Mixture Models. in *Encyclopedia of Biometrics* 659–663 (Springer, Boston, MA, 2009). doi:10.1007/978-0-387-73003-5_196.
23. Neath, A. A. & Cavanaugh, J. E. The Bayesian information criterion: background, derivation, and applications. *WIREs Comput Stat* **4**, 199–203 (2012).
24. Blum, M. G. B. & François, O. Which Random Processes Describe the Tree of Life? A

Large-Scale Study of Phylogenetic Tree Imbalance. *Syst. Biol.* **55**, 685–691 (2006).

25. Calinski, T. & Harabasz, J. A dendrite method for cluster analysis. *Commun. Stat. - Theory Methods* **3**, 1–27 (1974).
26. Rykov, A., De Amorim, R. C., Makarenkov, V. & Mirkin, B. Inertia-Based Indices to Determine the Number of Clusters in K-Means: An Experimental Evaluation. *IEEE Access* **12**, 11761–11773 (2024).
27. Rousseeuw, P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **20**, 53–65 (1987).
28. Farris, F. A. The Gini Index and Measures of Inequality. *Am. Math. Mon.* **117**, 851–864 (2010).
29. Hadfield, J. *et al.* Nextstrain: real-time tracking of pathogen evolution. *Bioinformatics* **34**, 4121–4123 (2018).
30. Kutschera, L. S. & Wolfinger, M. T. Evolutionary traits of Tick-borne encephalitis virus: Pervasive non-coding RNA structure conservation and molecular epidemiology. *Virus Evol.* **8**, veac051 (2022).
31. Dong, R., Pei, S., Yin, C., He, R. L. & Yau, S. S.-T. Analysis of the Hosts and Transmission Paths of SARS-CoV-2 in the COVID-19 Outbreak. *Genes* **11**, 637 (2020).
32. Hill, V. *et al.* A new lineage nomenclature to aid genomic surveillance of dengue virus. *PLOS Biol.* **22**, e3002834 (2024).
33. Happi, C. *et al.* Urgent need for a non-discriminatory and non-stigmatizing nomenclature for monkeypox virus. *PLOS Biol.* **20**, e3001769 (2022).
34. Parks, D. H. *et al.* GTDB: an ongoing census of bacterial and archaeal diversity through a phylogenetically consistent, rank normalized and complete genome-based taxonomy. *Nucleic Acids Res.* **50**, D785–D794 (2022).
35. Bielski, C. M. *et al.* Genome doubling shapes the evolution and prognosis of advanced

cancers. *Nat. Genet.* **50**, 1189–1195 (2018).

36. McPherson, A. *et al.* Ongoing genome doubling shapes evolvability and immunity in ovarian cancer. *Nature* 1–10 (2025) doi:10.1038/s41586-025-09240-3.
37. Kaufmann, T. L. *et al.* MEDICC2: whole-genome doubling aware copy-number phylogenies for cancer evolution. *Genome Biol.* **23**, 241 (2022).
38. Feng, Y. *et al.* Serial innovations by Asgard archaea shaped the DNA replication machinery of the early eukaryotic ancestor. *Nat. Ecol. Evol.* 1–13 (2025) doi:10.1038/s41559-025-02882-6.
39. Tamarit, D. *et al.* Description of *Asgardarchaeum abyssi* gen. nov. spec. nov., a novel species within the class *Asgardarchaeia* and phylum *Asgardarchaeota* in accordance with the SeqCode. *Syst. Appl. Microbiol.* **47**, 126525 (2024).
40. Zhou, Z., Liu, Y., Anantharaman, K. & Li, M. The expanding Asgard archaea invoke novel insights into Tree of Life and eukaryogenesis. *mLife* **1**, 374–381 (2022).
41. Stiller, J. *et al.* Complexity of avian evolution revealed by family-level genomes. *Nature* **629**, 851–860 (2024).
42. Kratter, A. W. The Howard and Moore Complete Checklist of the Birds of the World. *The Auk* **122**, 712–714 (2005).
43. Sangster, G. *et al.* Phylogenetic definitions for 25 higher-level clade names of birds. *Avian Res.* **13**, 100027 (2022).
44. Ohno, S. *Evolution by Gene Duplication*. (Springer, Berlin, Heidelberg, 1970). doi:10.1007/978-3-642-86659-3.
45. Innan, H. & Kondrashov, F. The evolution of gene duplications: classifying and distinguishing between models. *Nat. Rev. Genet.* **11**, 97–108 (2010).
46. Thomas, P. D. *et al.* PANTHER: Making genome-scale phylogenetics accessible to all. *Protein Sci.* **31**, 8–22 (2022).

47. Huminiecki, L. & Wolfe, K. H. Divergence of Spatial Gene Expression Profiles Following Species-Specific Gene Duplications in Human and Mouse. *Genome Res.* **14**, 1870–1879 (2004).

48. Cusack, B. P. & Wolfe, K. H. Not Born Equal: Increased Rate Asymmetry in Relocated and Retrotransposed Rodent Gene Duplicates. *Mol. Biol. Evol.* **24**, 679–686 (2007).

49. Pegueroles, C., Laurie, S. & Albà, M. M. Accelerated Evolution after Gene Duplication: A Time-Dependent Process Affecting Just One Copy. *Mol. Biol. Evol.* **30**, 1830–1842 (2013).

50. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *J. R. Stat. Soc. Ser. B Methodol.* **57**, 289–300 (1995).

51. Owen, N. R., Gumbs, R., Gray, C. L. & Faith, D. P. Global conservation of phylogenetic diversity captures more than just functional diversity. *Nat. Commun.* **10**, 859 (2019).

52. Grover, S., Markin, A., Anderson, T. K. & Eulenstein, O. Phylogenetic diversity statistics for all clades in a phylogeny. *Bioinformatics* **39**, i177–i184 (2023).

53. Faith, D. P. Conservation evaluation and phylogenetic diversity. *Biol. Conserv.* **61**, 1–10 (1992).

54. Matsen, F. A., Gallagher, A. & McCoy, C. O. Minimizing the Average Distance to a Closest Leaf in a Phylogenetic Tree. *Syst. Biol.* **62**, 824–836 (2013).

55. González-Orozco, C. E., Sosa, C. C., Thornhill, A. H. & Laffan, S. W. Phylogenetic diversity and conservation of crop wild relatives in Colombia. *Evol. Appl.* **14**, 2603–2617 (2021).

56. Simpson, E. H. Measurement of Diversity. *Nature* **163**, 688–688 (1949).

57. Piñeiro, C. & Pichel, J. C. Efficient phylogenetic tree inference for massive taxonomic datasets: harnessing the power of a server to analyze 1 million taxa. *GigaScience* **13**, giae055 (2024).

58. Huson, D. H. & Bryant, D. Application of Phylogenetic Networks in Evolutionary Studies. *Mol. Biol. Evol.* **23**, 254–267 (2006).

59. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).

60. de Vienne, D. M., Aguileta, G. & Ollier, S. Euclidean Nature of Phylogenetic Distance Matrices. *Syst. Biol.* **60**, 826–832 (2011).

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

62. Rosenberg, A. & Hirschberg, J. V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (ed. Eisner, J.) 410–420 (Association for Computational Linguistics, Prague, Czech Republic, 2007).

63. Hubert, L. & Arabie, P. Comparing partitions. *J. Classif.* **2**, 193–218 (1985).

64. Romano, S., Vinh, N. X., Bailey, J. & Verspoor, K. Adjusting for chance clustering comparison measures. *J Mach Learn Res* **17**, 4635–4666 (2016).

65. Kvålseth, T. O. On Normalized Mutual Information: Measure Derivations and Properties. *Entropy* **19**, 631 (2017).

66. M. Coronado, T., Mir, A., Rosselló, F. & Rotger, L. On Sackin's original proposal: the variance of the leaves' depths as a phylogenetic balance index. *BMC Bioinformatics* **21**, 154 (2020).

67. Stamatakis, A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics* **30**, 1312–1313 (2014).

Supplementary

Figures

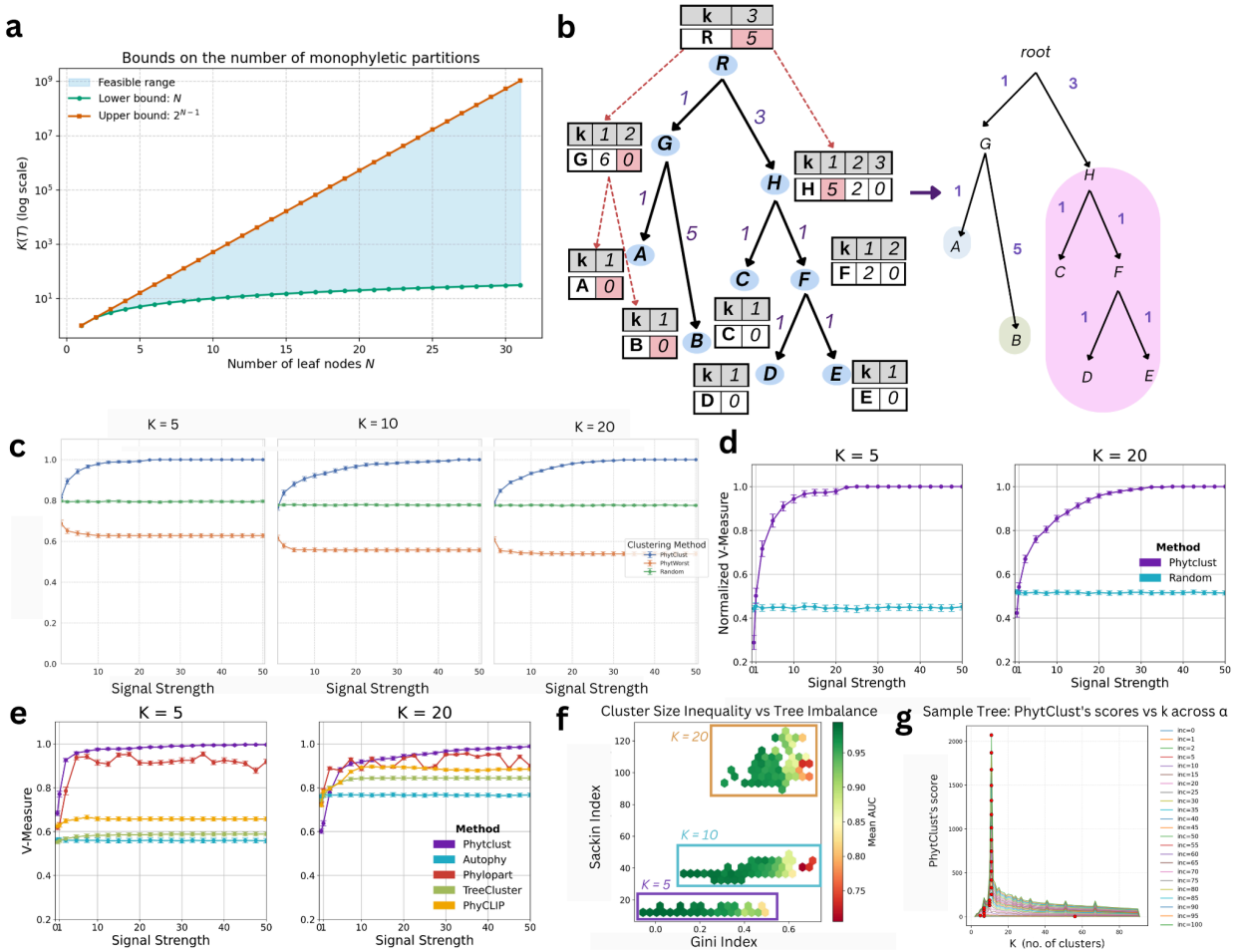
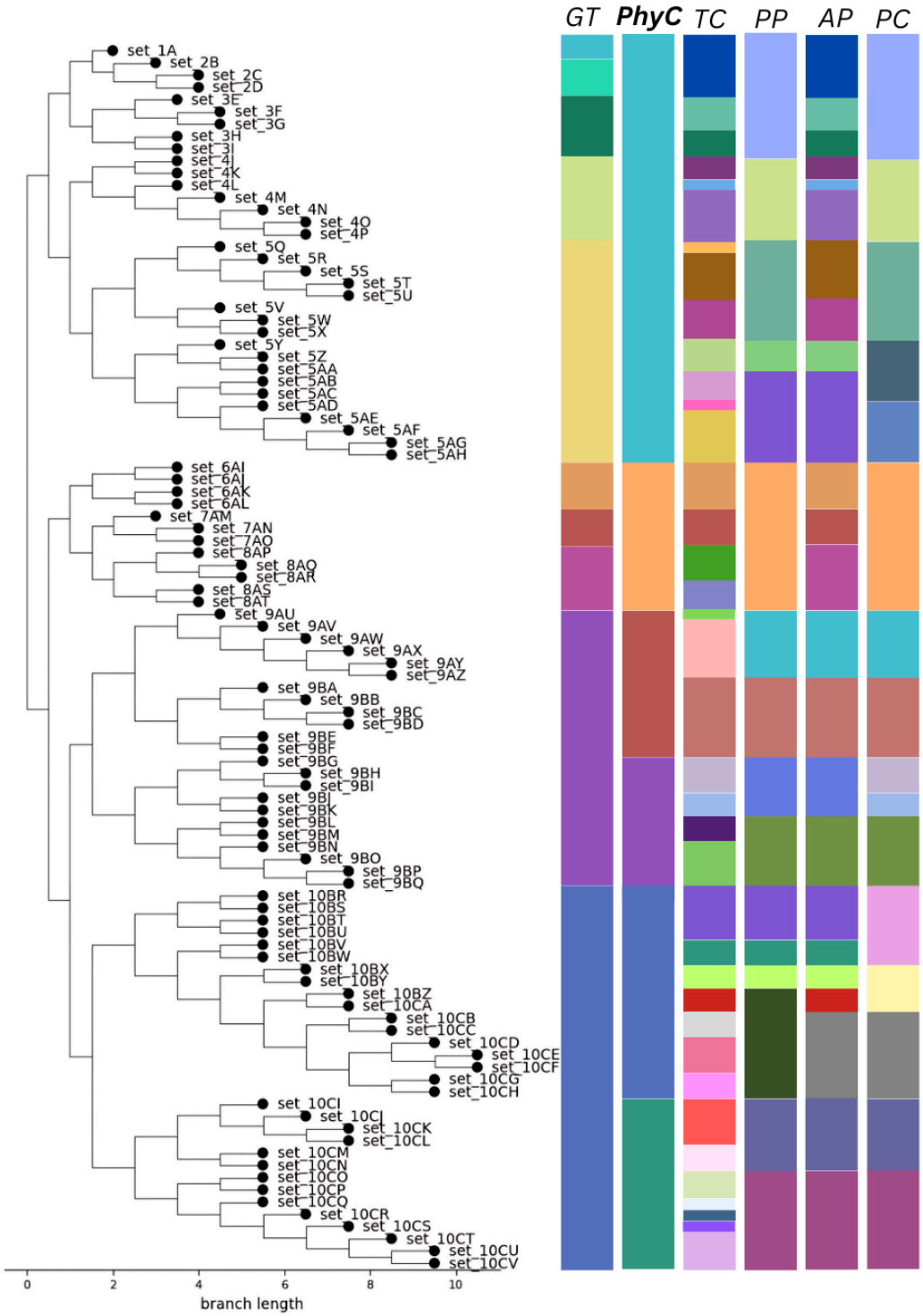
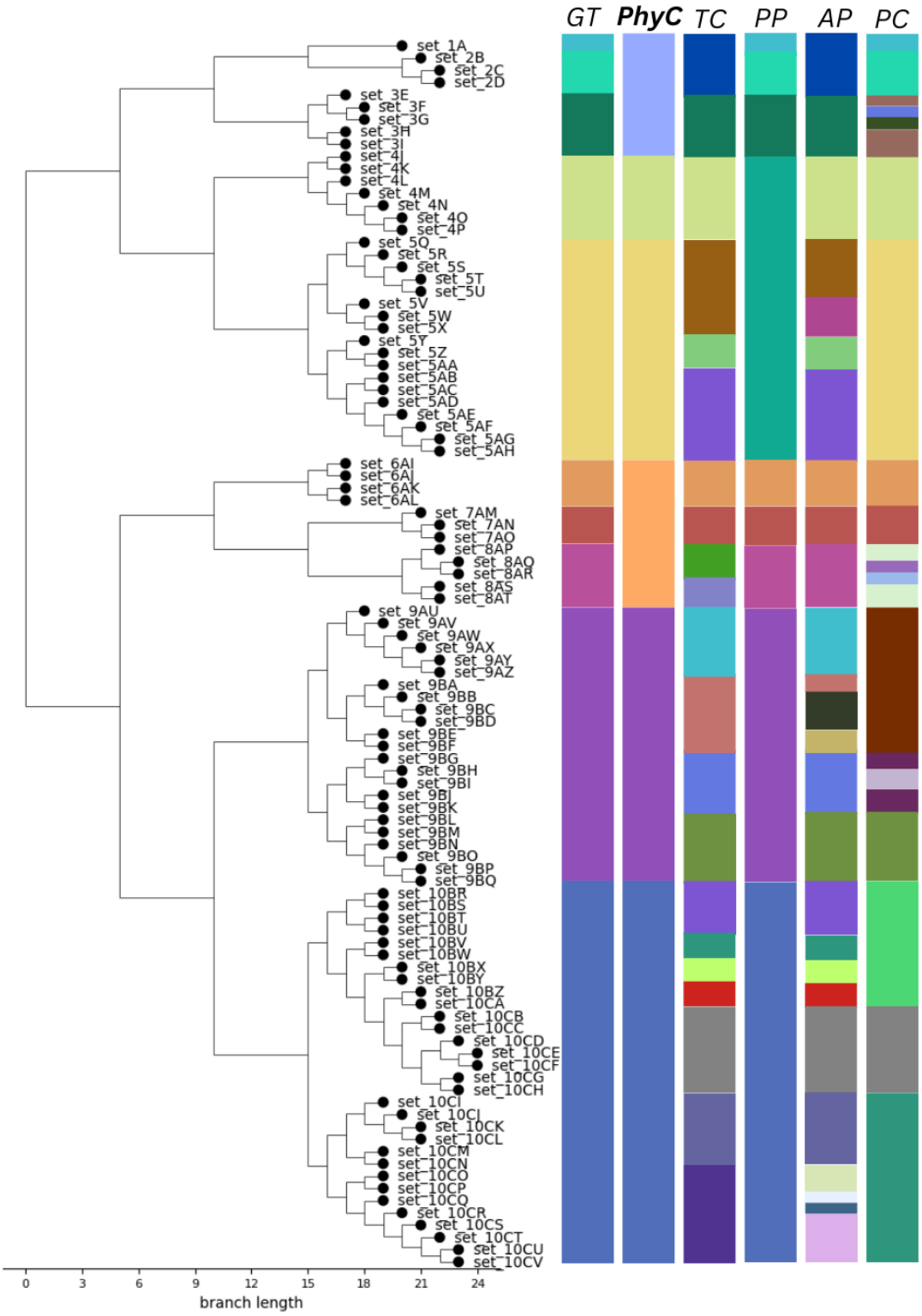


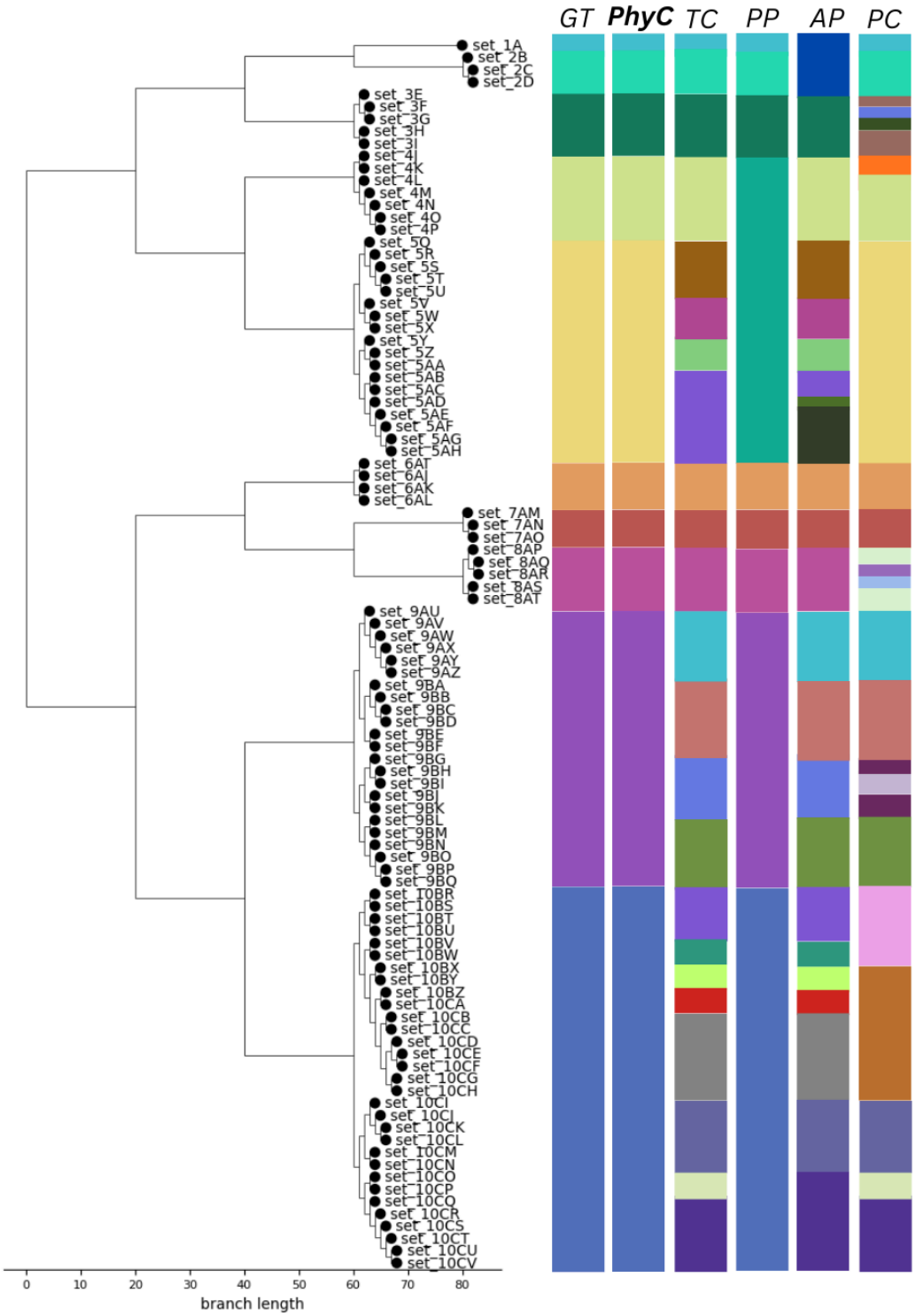
Fig. 1: (a) Number of possible combinations of monophyletic partitions on caterpillar trees and perfectly balanced trees. The shaded region indicates the feasible range of monophyletic partitions for trees that lie between these two extremes. (b) Example dynamic programming table for a sample tree. For k = 3, the entry at the root points to child H as one cluster and to child G as split into subclusters. The resulting optimal set of clusters that minimizes β is [A], [B], [C, D, E]. (c) Simulation results comparing PhytClust with random monophyletic partitions and with "PhytWorst," in which the optimization objective is reversed to maximize β(k) and intentionally produce poor clusters. (d) Simulation results comparing PhytClust with random monophyletic partitions for k = 5 and k = 20 on trees with N = 100 leaves. (e) Simulation results for k = 5 and k = 20 on trees with N = 100 leaves across the five phylogenetic clustering tools. (f) Cluster size inequality (Gini index) versus tree imbalance (Sackin index) for N = 100 and K = 5, 10, 20. Higher Gini index values correspond to worse performance of PhytClust. (g) Example tree showing the optimal k selected by PhytClust in red. As α increases, the peak at the target k* becomes progressively more pronounced.*



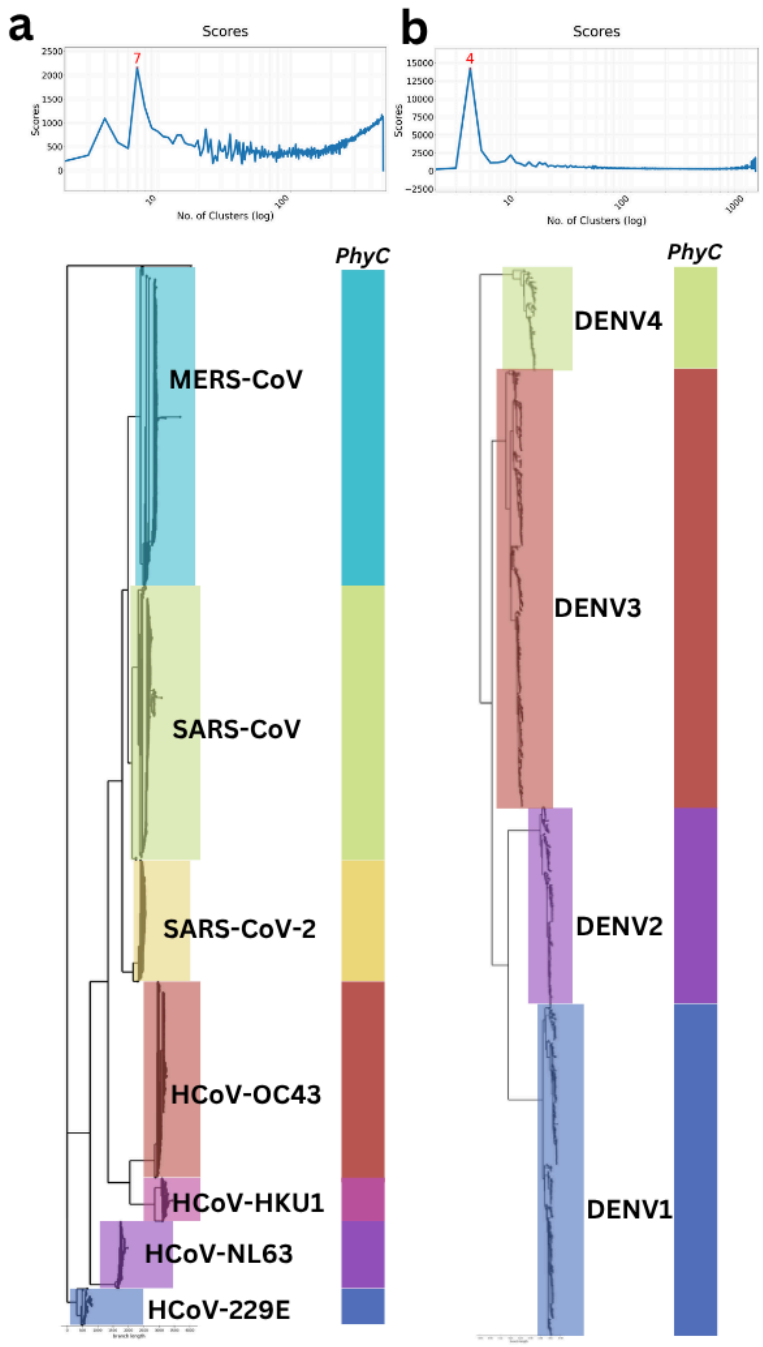
Supplementary Fig 2: Sample tree with target clusters (GT) (N=100, K=10) in comparison with clusters identified by PhytClust (PhyC), TreeCluster (TC), PhyloPart (PP), AutoPhy (AP) and PhyCLIP (PC) when α = 1



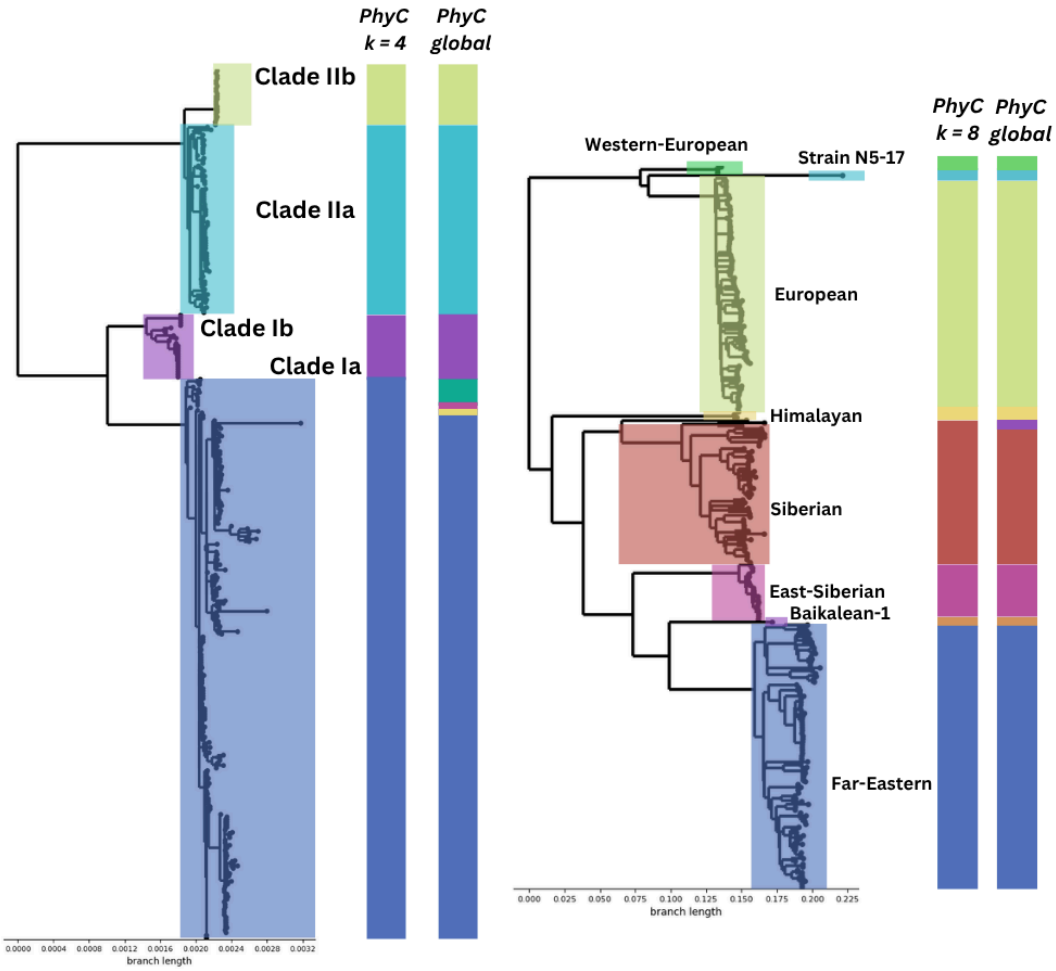
Supplementary Fig 3: Sample tree with target clusters (GT) (N=100, K=10) in comparison with clusters identified by PhytClust (PhyC), TreeCluster (TC), PhyloPart (PP), AutoPhy (AP) and PhyCLIP (PC) when α = 5



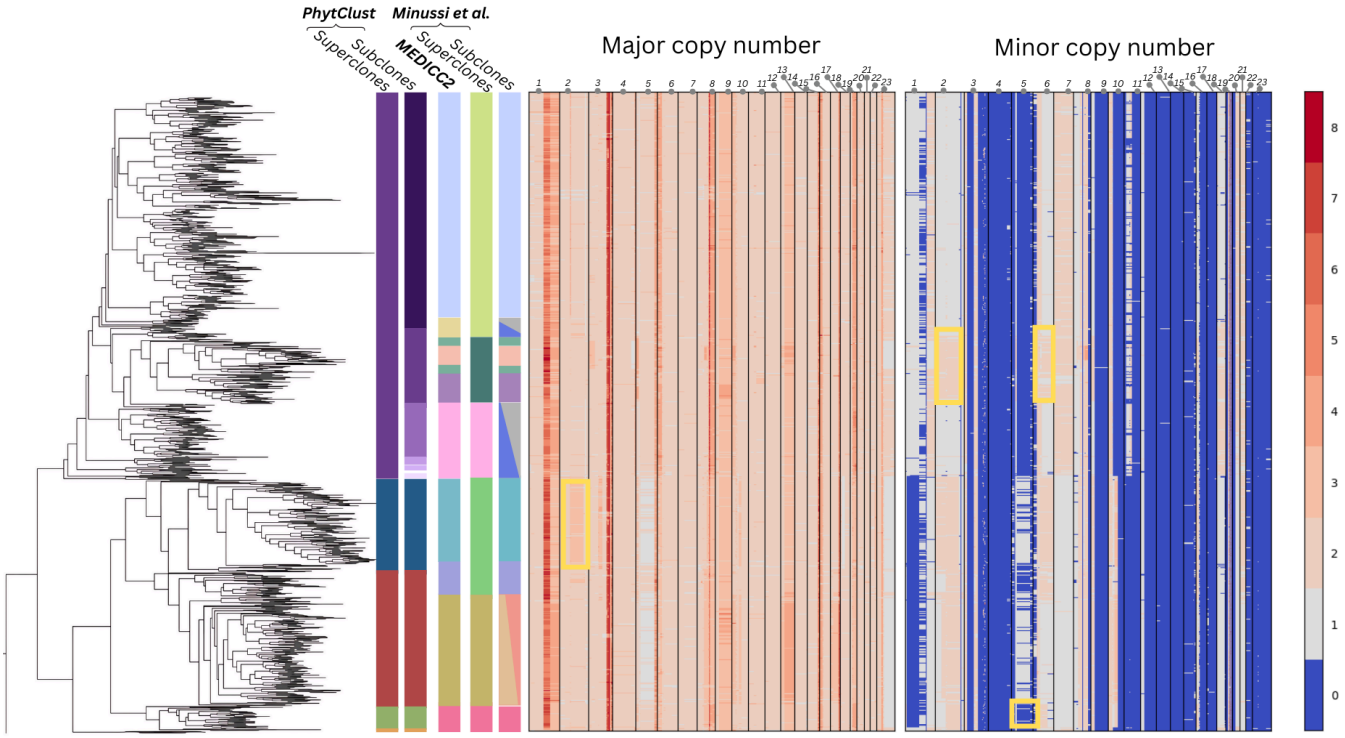
Supplementary Fig 4: Sample tree with target clusters (GT) (N=100, K=10) in comparison with clusters identified by PhytClust (PhyC), TreeCluster (TC), PhyloPart (PP), AutoPhy (AP) and PhyCLIP (PC) when α = 5



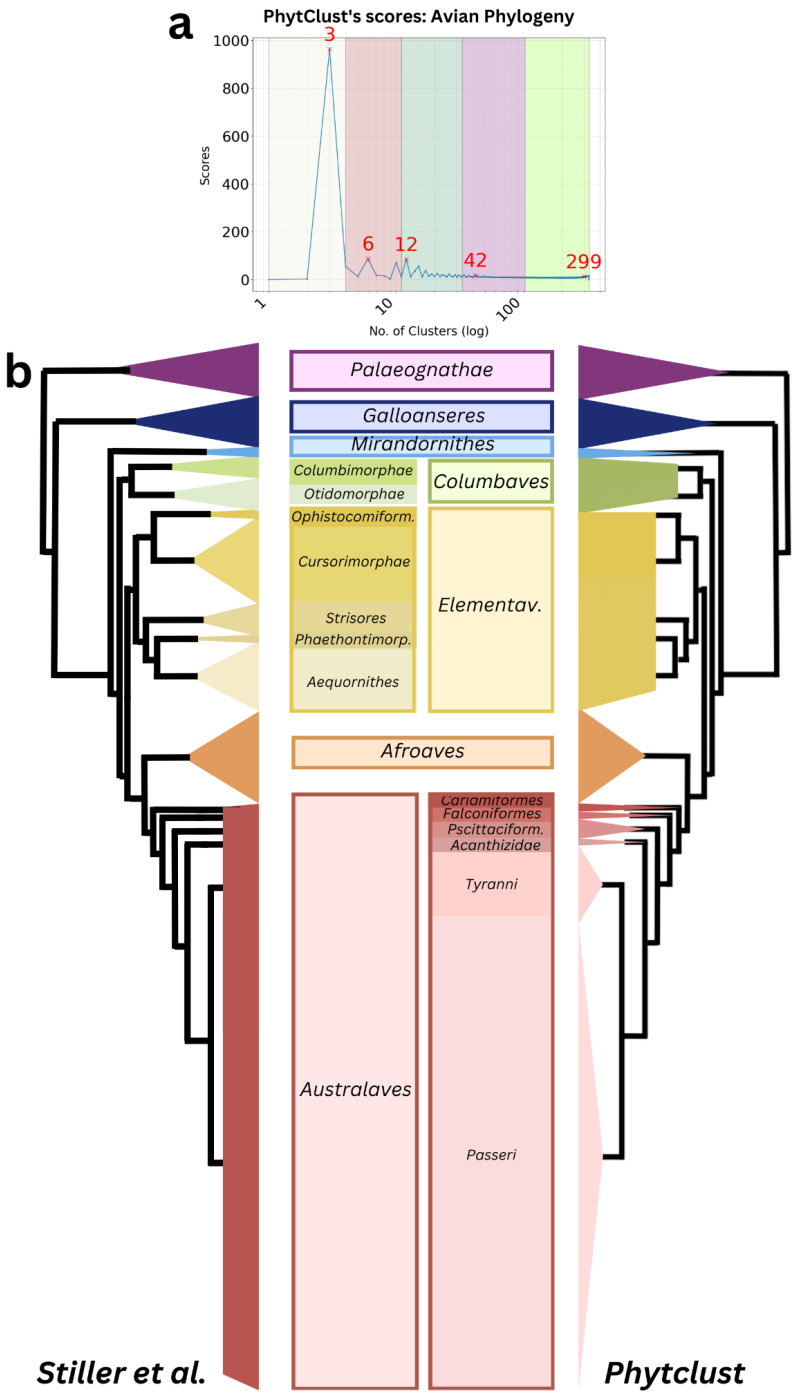
Supplementary Fig 5: a) Coronavirus phylogeny with annotated families as well as PhytClust's chosen solution (k = 7) and PhytClust's scores; b) Dengue phylogeny with annotated Nextstrain clades as well as PhytClust's chosen solution (k* = 4) and PhytClust's scores.*



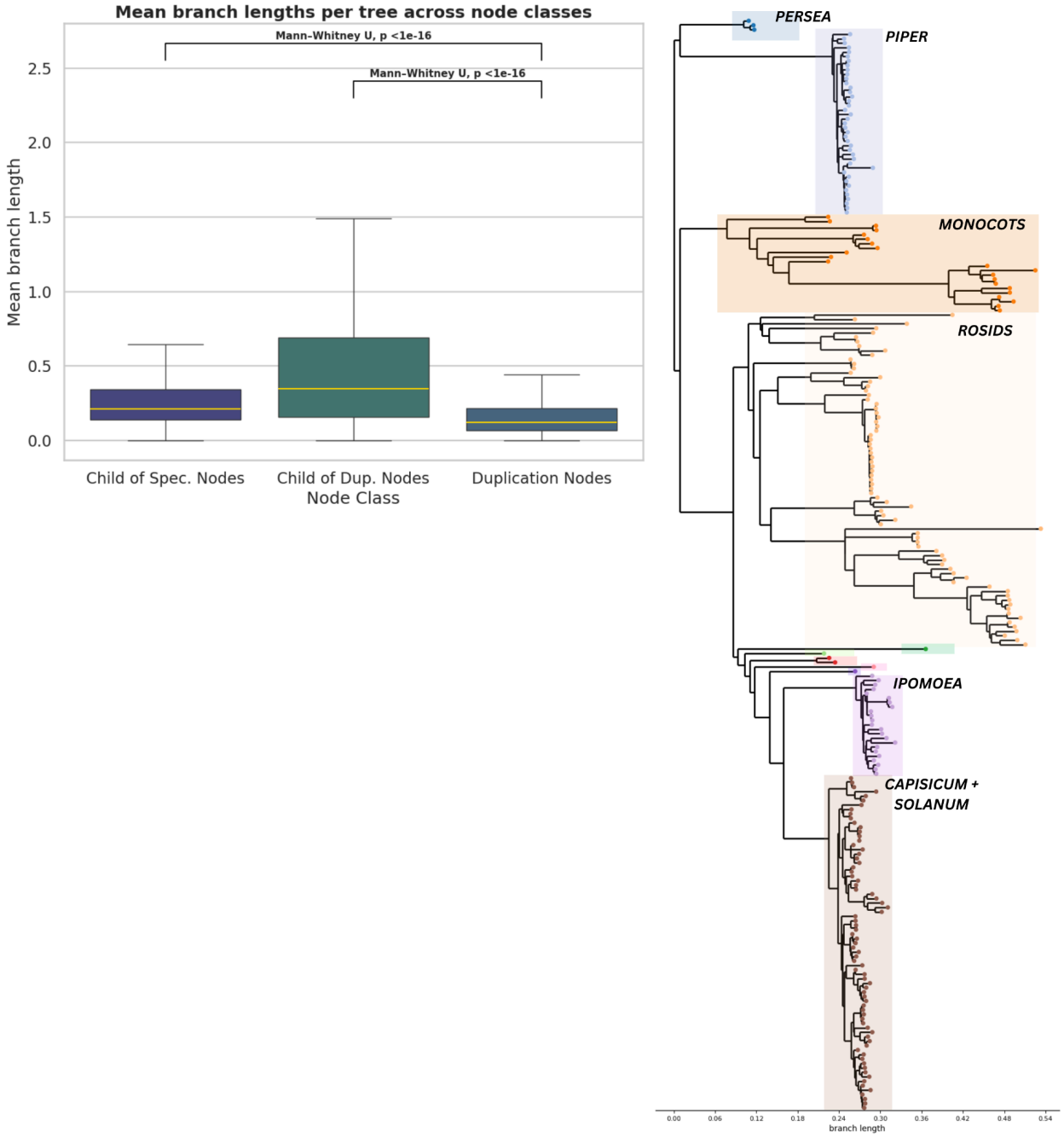
Supplementary Fig 6: a) Monkeypox phylogeny with annotated clades as well as PhytClust's chosen solution (k = 7) and PhytClust's solution for k = 4; b) Tick-borne encephalitis viral phylogeny with annotated Nextstrain clades as well as PhytClust's chosen solution (k* = 9) and PhytClust's solution for k = 8.*



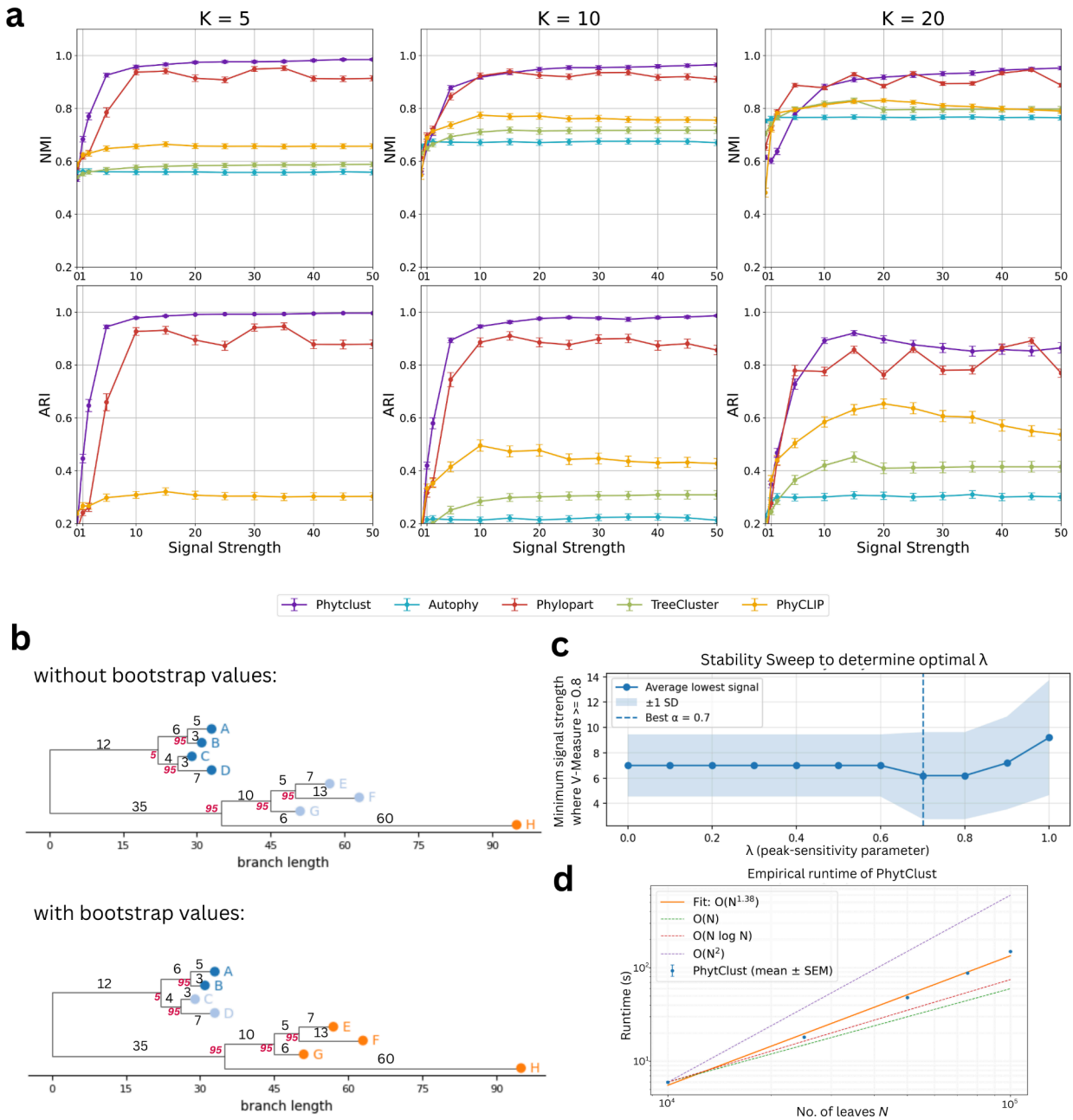
Supplementary Fig 7: a) Minussi TN-2 phylogeny with original subclones and superclones from Minussi et al, subclones annotated by MEDICC2 authors as well as superclones and subclones identified by PhytClust.



Supplementary Fig 8: (a) PhytClust's scores for the avian phylogeny obtained from Stiller et al and stratified into 5 Clade Levels. (b) Comparison of clade annotations from Stiller et al and PhytClust's CL3 clusters.



Supplementary Fig. 9: (a) distribution of z-normalised branch lengths of the phylogenetic trees from PANTHER database for all branches leading to a duplication node and branches following a duplication or speciation node.(b) Complete phylogenetic tree showing PhytClust cluster assignments, together with the most appropriate taxonomic annotations for each cluster



Supplementary Fig.10: (a) Simulation results reporting Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) for PhytClust, AutoPhy, PhyCLIP, TreeCluster and PhyloPart of trees with N =100, K = 10. (b) Sample tree showing differences in optimal clusters k chosen by PhytClust with and without accounting for bootstrap values (c) Stability sweep over various values of Lambda using simulated trees. (d) Empirical runtime scaling of PhytClust.*

Pseudocode for Dynamic Programming:

```
function compute_DP_table(tree, num_leaves_per_node):
      input tree: tree without outgroup (if given)
      input num_leaves_per_node: dictionary where keys are nodes and
      values are the number of leaves under each node


      postorder_nodes <- list of all nodes in tree for postorder
      traversal
      num_node <- length of postorder_nodes

      dp_table <- [None] * num_nodes
      backtrack <- [None] * num_nodes

      for each node in postorder_nodes do
            num_leaves_node <- num_leaves_per_node[node]
            if node is leaf node:
                  dp_table[node] <- 0.0
                  backtrack[node] <- 0.0
            left, right <- children of node
            left_dp <- dp_table[left]
            right_dp <- dp_table[right]

            cost_node_as_one_cluster <- left_dp[0] + left_child
            branch_length * left_child_num_leaves + right_dp[0] +
            right_child_branch_length * right_child_num_leaves

            if use_support_value do
                  cost_node_as_one_cluster /= node_support_value / 100

            dp_array[0] <- cost_one_cluster
            backtrack_array[0, 0] <- 0
            backtrack_array[1, 0] <- 0

            limit <- num_leaves_node + 1
            for each cluster_size in range(1, limit) do
                  valid_i <- np.arange(cluster_size)

                  valid_filter_i <- valid_i < len(left_dp)
                  valid_i <- valid_i[valid_filter_i ]
                  valid_j <- cluster_size - 1 - valid_i

                  valid_filter_j <- valid_j < len(right_dp)
                  possible_i <- valid_i[valid_filter_i]
                  possible_j <- valid_j[valid_filter_j]
```

```
        if valid_i.size > 0 do
        left_dp_values <- left_dp[valid_i]
        right_dp_values <- right_dp[valid_j]
              sub_scores <- left_dp_values + right_dp_values
              min_index <- argmin(sub_scores)
              min_score <- sub_scores[min_index]
              dp_array[cluster_size] <- min_score
              backtrack_array[0, cluster_size] <-
              valid_i[min_index]
              backtrack_array[1, cluster_size] <-
              possible_j[min_index]

dp_table[node_id] <- dp_array
backtrack[node_id] <- backtrack_array
dp_table[left_id] <- None
dp_table[right_id] <- None
```

Proof A: Relationship Between the Dynamic Programming (β) Score and the Calinski–Harabasz Index

Aim: Given a phylogenetic tree with *n* terminal nodes, find a score that when minimized, gives the most suitable clustering sets for *k* monophyletic clusters. The DP score for a phylogenetic tree is given by:

$$\beta(c_k) = \sum_{c_i \in C_k} \sum_{i \in c_i} d(i, MRCA(i)) \qquad (1)$$

Where,

C_i is a given cluster in the set of clusters C_k

C_k is any set of clusters possible for *k* monophyletic clusters

MRCA(i) is the MRCA of the set of terminal nodes in a cluster

The Calinski Harabasz score is given by

$$\mathrm{CH}(c_k) = \frac{\mathrm{BSS}}{\mathrm{WSS}} \mathrm{X} \frac{\mathrm{N\text{-}K}}{\mathrm{K\text{-}1}} \qquad (2)$$

Where,

TSS is the Total Sum of Squares

BSS is the between cluster Sum of Squares

WSS is the within cluster Sum of Squares

N is the number of data points

K is the number of clusters, for Euclidean distances

Since TSS = WSS + BSS,

$$\mathrm{CH}(c_i) = \frac{\text{TSS-WSS}}{\text{WSS}} \text{X } m$$

where,

$$m = \frac{N-K}{K-1}$$

$$TSS = \sum_{i=1}^{N} (x_i - X)^2$$

$$WSS = \sum_{k=1}^{K} \sum_{c_i \,\epsilon\, C_k} ||x_i - C_k||^2$$

x_i are data points in a cluster, X is the centroid of a cluster.

Patristic distances in a phylogenetic tree are equivalent to squared Euclidean distances, as proven by de Vienne et al[60]. If we consider the MRCA of terminal nodes in a cluster to be the local centroid and the root to be the global centroid:

$$\beta(1) = \sum_{c_i \,\epsilon\, C_k} \sum_{i \,\epsilon\, c_i} d(i, \rho) = TSS$$

Where ρ is the root of the phylogenetic tree.

Similarly,

$$\beta(c_k) = \sum_{c_i \,\epsilon\, C_k} \sum_{i \,\epsilon\, c_i} d(i, MRCA(i)) = WSS$$

The Calinski Harbasz score can now be rewritten as:

$$\mathrm{CH}(c_k) = \frac{\beta(1) - \beta(c_k)}{\beta(c_k)} \mathrm{X}\ m$$

∴

$$1/\mathrm{CH}(c_k) = \frac{\beta(c_k)}{m(\beta(1) - \beta(c_k))}$$

As $\mathrm{CH}(c_k)$ increases, $\beta(c_k)$ decreases. If we consider $\mathrm{CH}(c_k) = y$,.$\beta(1)$ = a, $\beta(c_k)$ = x, we can rewrite equation as:

$$\frac{1}{x} = \frac{y}{m(x-a)}$$

$$1 = y\left(\frac{x}{m}\right) - y\left(\frac{x^2}{m}\right)$$

Hence, the relationship between both scores is described by a quadratic function. The graph for this relationship would be parabolic in nature.