IJITCE

ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

Email : ijitce.editor@gmail.com or editor@ijitce.com

International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021

Biology Sequence Clustering Via Phylogenetic Trees

ANSHUMAN MISHRA

Abstract

For many bioinformatics tasks, it is necessary to group comparable sequences together. Sequences tend to group together because of the evolutionary links among them. And yet, despite this evidence and the obvious ways in which a

Despite the fact that a phylogenetic tree may be used to create groups, most sequence clustering tools instead employ pairwise sequence distances to do their analyses. We contend that tree-based clustering is not being fully used because of the development of large-scale phylogenetic inference. For each given tree, we describe a class of optimization problems that, when solved, provide the fewest possible clusters while satisfying specified heterogeneity requirements. We focus on three distinct restrictions, which limit either (1) the size of each cluster, (2) the total length of its branches, or (3) the length of chains of pairwise distances. For two of the three requirements, the methods have been known for some time in the theoretical computer science literature. The time required to solve these issues grows linearly with the size of the tree. Using these techniques, we develop a program called TreeCluster and evaluate it on three different uses: clustering of OTUs in microbiome data, clustering of HIV transmission, and divide-and-conquer multiple sequence alignment. We demonstrate how TreeCluster's use of tree-based distances produces more internally consistent clusters than competing methods and boosts the efficiency of subsequent applications. Check out https://github.com/niemasd/TreeCluster to download TreeCluster.

Introduction

Homologous molecular sequences may exhibit striking resemblance across species and even within the same genome owing to their common evolutionary history. Due of these commonalities, several applications have begun by classifying a wide variety of sequences into high-similarity sequence sets clustered for further processing. Clusters have varying, context-specific meanings. Operational Taxonomic Units (OTUs) are used as part of the standard process for evaluating 16S microbiome data [1-3]. These are groups of closely related sequences that do not deviate beyond a particular threshold.

One such topic where the standard method is to infer causality is HIV transmission inference cluster HIV sequences from various people depending on how similar they are to one another (again, using a threshold), and then utilize these clusters as surrogates for disease transmission hotspots [4, 5].

Homologous similarities stem from their shared evolutionary histories.

ASSISTANT PROFESSOR, Mtech,
Department of CSE
Gandhi Institute for Technology,Bhubaneswar.

Phylogenetic trees are a useful tool for displaying sequence data. Recent approaches can estimate approximate maximum-likelihood (ML) phylogenetic trees in sub-quadratic time, allowing them to scale to datasets of even millions of sequences [8]. This allows the phylogenetic tree to be inferred from sequence data [6, 7]. In addition, utilizing divide-and-conquer techniques [9, 10], precise alignment of datasets including hundreds of thousands of species is now feasible (a precondition to most phylogenetic reconstruction approaches).

Current sequence clustering algorithms typically accept as input pairwise distances between sequences but do not make use of phylogenetic trees. The popular UCLUST [2] algorithm, for instance, looks for a clustering that maximizes the Hamming distance between cluster centers while minimizing the Hamming distance of sequences to the cluster centroid. Several alternative clustering approaches, such as those for large protein sequence databases [13] and gene family circumscription [11, 12], have been developed for specific applications.

There are two possible benefits of using phylogenies for clustering. I Phylogeny-based grouping has the ability to represent not just evolutionary distances (i.e., branch lengths) but also connections (i.e., the tree topology), since phylogenies explicitly strive to infer the evolutionary past. Keep in mind that the lengths of the branches in a phylogeny are statistically rigorous "corrections" of sequence distances based on a model [7, 14], and therefore they may more accurately represent the degree of divergence between the species. ii) The tree may be inferred using subquadratic techniques, which can enhance performance and scalability by removing the requirement to calculate all pairwise distances.

Furthermore, a phylogeny is frequently easily accessible since it must be inferred for reasons other than clustering. Despite these possibilities, however, no known systematic approach to phylogeny-guided grouping has been developed. ClusterPicker [15] was developed for investigating HIV transmissions; it groups sequences based on distances using the phylogenetic tree as a constraint; nevertheless, it still use sequence distances rather than tree distances and scales cubically with respect to the number of sequences.

In the case of an ultrametric tree, in which the distances between nodes on the tree are all the same, it is straightforward to use the tree to cluster sequences based on their evolutionary relationships (Fig 1A). By roots the tree at its singular midway and continuing as previously, this method naturally applies to bare ultrametric trees as well. However, phylogenetic trees that are derived from data are seldom ultrametric. Inferred trees may not be ultrametric even if the real tree is ultrametric, since different creatures might develop at different rates of evolution. When working with a non-ultrametric (and maybe unrooted) tree, it might be difficult to determine how to effectively cluster sequences (Fig 1B).

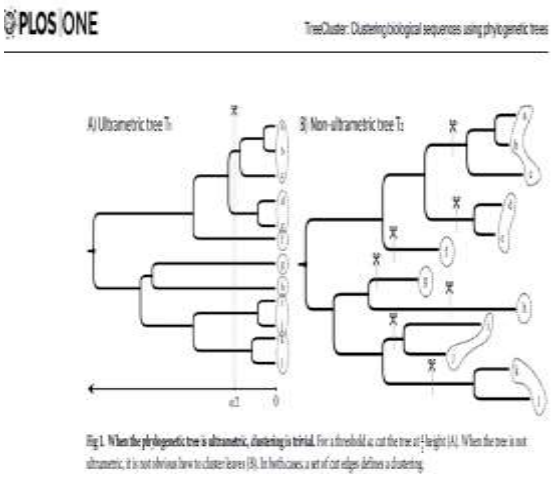
Tree-based clustering may be approached in a number of ways, one of which is as an optimization issue. A common style of issue definition goes like this: "discover the minimal number of clusters such that specified criteria restrict each cluster." Interestingly, the theoretical computer science community has tackled at least two variants of similar optimization issues since the 1970s, often in the context of proving more difficult theorems. The tree partitioning issue entails slicing a tree into the least number of subtrees possible, with the goal of satisfying a certain bound on either the maximum route length between two nodes in the same subtree [17] or the total weight of all edges in each subtree [18]. Although simple linear-time methods exist for solving these issues perfectly, it is our impression that bioinformaticians mostly disregard them.

In this paper, we propose a quick and efficient tree-based clustering technique as a solution to a number of bioinformatics problems. In this work, a new class of tree-partitioning algorithms is presented.

International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021



difficulties and provide linear-time descriptions of solutions for three examples (two of which match to the aforementioned max and sum problems with known methods). In the next section, we demonstrate how tree-based grouping might enhance subsequent biological investigations in three distinct applications: operational taxonomic unit (OTU) definition for microbes, cluster analysis of HIV transmission, and divide-and-conquer multiple sequence alignment.

Substances and Techniques

Algorithms

Labeling the issue at hand. Consider the unrooted binary tree T = (V, E), which is a directed acyclic network with vertices V (of either degree 1 or 3), weighted edges E, and a leafset L V. It is common to abbreviate the notation for the distance between two leaves u and v on T to dT (u, v), however where it is not obvious, we will use the shorter notation d(u, v). An edge's "weight" (i.e., its "branch length") is expressed by the symbol "w" when the pair of vertices (u, v) is considered (u, v).

By severing a certain set of edges C E, we may define a clustering of the tree's leaves, or T.

If we remove an edge set C from E and assign the leaves of each of the resulting linked components to a set Li (note: N |C| + 1), then we have a partition L1, L2, LN of L that is an acceptable clustering.

Let fT: 2L! R represent a function from a subset of the leafset L to a real integer, and assume that tree T is a tree. fT is often specified as a function of the edge weights in a cluster, and its goal is to quantify the variety of elements at the leaves inside each cluster. A good example is the diameter of a subset: fT = maxu,v2L dT. (u, v). We construct a class of tasks with the objective of reducing the total number of clusters while imposing fT-defined restrictions on each cluster. In formal terms:

Instance 1: A Definition (Min-cut partitioning problem family). Find the smallest cardinality (N) acceptable partition L1...LN of L satisfying 8i, fT for a given tree T with leafset L and a real integer.

Requiring all pairwise distances inside a cluster to be less than a specified threshold is a logical technique to reduce variety within the cluster.

As a second definition (Max-diameter min-cut partitioning problem). When fT = max u;v2L d u; v, the Min-cut partitioning issue (Definition 1) is referred to as the Max-diameter min-cut partitioning problem.

Possible drawbacks of max-diameter-min-cut partitioning include its sensitivity to outliers (the maximum distance inside a cluster is not always representative of the true distance between nodes).

the cluster's level of variety. Here's an obvious option that might help control for extreme cases:

Third Definition (Sum-length min-cut partitioning problem). To solve the Min-cut partitioning issue where T|L is the tree T limited to a subset of leaves L, and fT 14 X u;v2edgesTjL wu; v. This problem, which is also known as the Sum-length Min-Cut Partitioning Problem, is NP-complete.

In addition, we investigate a third issue, the significance of which we shall explain in detail:

4. Defined (Single-linkage min-cut partitioning problem). When fT = max S L f min u2S;v2L S du; vg, we refer to the Min-cut partitioning issue as a Single-linkage min-cut partitioning problem.

Next, we will demonstrate algorithms that solve the Max-diameter, Sum-length, and Singlelinkage min-cut partitioning issues in linear time. Two of the three algorithms, max and sum, have previously been defined in the theoretical computer science literature, and all three employ versions of the same greedy algorithm. However, we restate the answers

in the same terms and provide different justifications for their validity.

The Max-diameter min-cut problem can be solved in linear time. Initially, Parley et al. [17] provided a linear-time solution (with all edge weights equal to 1) to the Max-diameter min-cut partitioning issue. We propose a different argument for Algorithm 1, which is similar to the one given by Parley et al. (with the addition of branch lengths). To is an arbitrary rooting of T at o, and it is this form that the algorithm acts on. The subtree with root u is referred to as U. Let's name U's offspring Ul and Ur and their family tree Ul and Ur. Whenever it is obvious, we write wl for w(u, ul) and wr for w(u, ur).

First Algorithm: Time-consistent linear solution to Partitioning by maximum diameter minimum cut

Input: A tree For a given value of (Voltage, Energy), we need to meet a threshold of 1 B(v) 0 in order for v 2 V 2 to equal (U, E). Internal node traversal in post-order 2, If B(ul) + wl + B(ur) + wr >, go to step 4, and if B(ul) + wl B(ur) + wr >, proceed to step 3.

We characterize B(C, u) as the distance from u to the most distant linked leaf in U in the clustering specified by C, for a cut set C of the tree. The algorithm does a bottom-up tree traverse, and at each node u it encounters, it has the option of severing one of the child edges leading to it.

Because of this, we create a clustering current Cu at each step and denote it by the abbreviation B(u) (Cu, u).

Once we reach u, additional connections between the two trees Ur and Ul will emerge. The longest of such routes is B(ul) + wl + B(ur) + wr in length. When this number is more than the threshold, we break either (u, ur) or (u, ul), whichever results in the smallest value for B. (u). To emphasize, B(u) is always well-defined since the method never cuts more than one child edge from any node.

Assume for the sake of this theorem that the goal function, denoted by A(o), is the least number of clusters under U with a diameter smaller than. The first algorithm finds the clustering that minimizes A(o) given the tree To as its root. Furthermore, the algorithm selects arg minC B from all feasible clusterings of this kind (C, o).

Let C0 be the cut set generated from Algorithm 1 when applied to a tree T with any rooted To. The partitioning issue with Max-diameter min-cut is ideally solved by set C0.

Both the theorem's proof and the corollary's proof may be found in the S1 Appendix.

Partitioning issue with minimum sum length has a linear solution. Kundu et al. [18] previously described a linear-time approach that divides trees into clusters with total node weights less than or equal to. To get an answer to the Sumlength

To solve the min-cut partitioning issue, we provide a modified version of the original technique that operates on edge (rather than node) weights and is dedicated to binary trees. The Sum-length min-cut partitioning issue is best solved by Algorithm 1 with only two tweaks (see Algorithm A in S1 Appendix). As a first tweak, we use the new auxiliary variable B(C, u) to represent the aggregate of edge weights for all ancestors of u at the current processing step of the algorithm. Two, let's say we're doing a bottom-up traversal of the internal nodes of To, and for node u, w.l.o.g, we're going to allow B(ul) + wl B(ur) + wr). We sever the edge if the total of the combined subtree's branch lengths is more than (u, ul). When calculating B(u), this algorithm uses a different formula from the one found in Algorithm 1: B(ul) + wl + B(ur) + wr. The accuracy of the algorithm is shown in S1 Appendix, using a pattern similar to that of the proof for Algorithm 1.

Minimal cut tree partitioning with a single link. The issue of Single-linkage min-cut partitioning (Definition 4) is next, and it may be seen as a relaxation of the Max-diameter min-cut partitioning. The following definition will help explain why this is an issue.

5. Defined (Single-linkage clustering). If there exists a chain H 14 c0; c1;... ; cm; cm1, with a = c0 and b = cm+1 and for every 0 I m, we get d(ci, ci+1), then we say that the partition of L is a Single-linkage clustering.

The result is that pairs of nodes are always placed in the same cluster if their distance is less than the

International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021

cutoff value (the rest follows from transitivity). Determination 4's fT is supported by the following result (proven in S1 Appendix).

Optimal solutions to the Single-linkage min-cut partitioning issue (Definition 4) and the Single-linkage clustering (Definition 5) are equivalent, as stated in Proposition 1.

The Single-linkage min-cut partitioning issue may be solved in linear time, as shown by Algorithm 2.

For each node u, the method uses post-order traversal to locate the nearest leaf in the left and right sub-trees of u, and then uses pre-order traversal to locate the nearest leaf outside the sub-tree rooted at u. Then, it does a post-order traversal, cutting each child edge if the shortest path between the node and any leaf outside the node is longer than the threshold distance. The following theorem (with proof in S1 Appendix) establishes the algorithm's soundness.

Procedure 2: SINGLE-LINKAGE Partitioning with a minimal cut distance using a single link

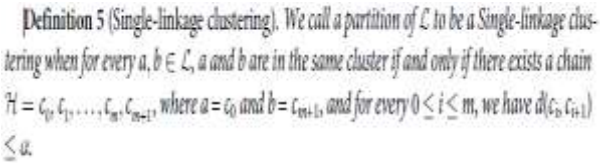
Lower than [u] minimum

Above[u] Postorder recursion of (1, v, u2) for If you're in L, it'll take you 4 minutes to finish Step 3. 0; 5; 6; 6 min To the sub-minimum Under[ul] plus wl, minimum Below[ur] + wr); 7 for u 2 a sequential walk through the tasks on your To Do list; 8 if u 614 o, then 9 minutes To a degree greater than or equal to the minimum The minimum value is less than [below[s]] plus [w(v, s)]. For u 2, the post-order traversal of the internal nodes of For 11 if minimum Subsequent to[ul]+(wl)+(min) In the range [ur] - wr > and min Subsequent to[ul]+(wl)+(min) 12 E E(u, ul) 13 if min Above[u]> then Lower than [ul] + (wl) + (min) Given that [ur] + wr > and min Less than[ur] + wr + minimum If min [u]>above 14 E E(u, ur) 15 if min [u]>above Subsequent to[ul]+(wl)+(min) In the range [u] > and min The combination of "below" (ur) and "wr" (western) and "minimum" If (Above[u]) > (Tree), Phylogenetic tree-based clustering of biological sequences

Partitioning issue with minimum sum length has a linear solution. Kundu et al. [18] previously described a linear-time approach that divides trees into clusters with total node weights less than or equal to. To get an answer to the Sumlength

To solve the min-cut partitioning issue, we provide a modified version of the original technique that operates on edge (rather than node) weights and is dedicated to binary trees. The Sum-length min-cut partitioning issue is best solved by Algorithm 1 with only two tweaks (see Algorithm A in S1 Appendix). As a first tweak, we use the new auxiliary variable B(C, u) to represent the aggregate of edge weights for all ancestors of u at the current processing step of the algorithm. Two, let's say we're doing a bottom-up traversal of the internal nodes of To, and for node u, w.l.o.g, we're going to allow B(ul) + wl B(ur) + wr). We sever the edge if the total of the combined subtree's branch lengths is more than (u, ul). When calculating B(u), this algorithm uses a different formula from the one found in Algorithm 1: B(ul) + wl + B(ur) + wr. The accuracy of the algorithm is shown in S1 Appendix, using a pattern similar to that of the proof for Algorithm 1.

Minimal cut tree partitioning with a single link. The issue of Single-linkage min-cut partitioning (Definition 4) is next, and it may be seen as a relaxation of the Max-diameter min-cut partitioning. The following definition will help explain why this is an issue.


Definition 5 (Single-linkage clustering). *We call a partition of $\mathcal{L}$ to be a Single-linkage clustering when for every $a, b \in \mathcal{L}$, a and b are in the same cluster if and only if there exists a chain $\mathcal{H} = c_0, c_1, \ldots, c_m, c_{m+1}$, where $a = c_0$ and $b = c_{m+1}$, and for every $0 \leq i \leq m$, we have $dl(c_i, c_{i+1}) \leq u$.*

The result is that pairs of nodes are always placed in the same cluster if their distance is less than the cutoff value (the rest follows from transitivity). Determination 4's fT is supported by the following result (proven in S1 Appendix).

Optimal solutions to the Single-linkage min-cut partitioning issue (Definition 4) and the Single-linkage clustering (Definition 5) are equivalent, as stated in Proposition 1.

The Single-linkage min-cut partitioning issue may be solved in linear time, as shown by Algorithm 2.

For each node u, the method uses post-order traversal to locate the nearest leaf in the left and

International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021

right sub-trees of u, and then uses pre-order traversal to locate the nearest leaf outside the sub-tree rooted at u. Then, it does a post-order traversal, cutting each child edge if the shortest path between the node and any leaf outside the node is longer than the threshold distance. The following theorem (with proof in S1 Appendix) establishes the algorithm's soundness.

Algorithm 2: Single-Linkage Single-linkage min-cut partitioning

```
minBelow[u] ← minAbove[u] ← ∞ for v ← V
for u ∈ post order traversal of T° do
  if u in L then
    minBelow[u] ← 0;
  else
    minBelow[u] ← min(minBelow[u1] + w1, minBelow[u2] + w2);
for u ∈ pre order traversal of T° do
  if u ≠ o then
    minAbove[u] ← min(minBelow[s] + w(v, s), minAbove[v] + w(v, v));
for u ∈ post order traversal of internal nodes of T° do
  if minBelow[u1] + w1 + minBelow[u2] + w2 > α and
    minBelow[u1] + w1 + minAbove[u]> α then
    E ← E\{u, u1}
  if minBelow[u1] + w1 + minBelow[u2] + w2 > α and
    minBelow[u2] + w2 + minAbove[u]>α then
    E ← E\{u, u2}
  if minBelow[u1] + w1 + minAbove[u] > α and
    minBelow[u2] + w2 + minAbove[u] > α then
    E ← E\{v, u}
return Leafsets of every connected component in T°
```

The ideal solution to the Single-linkage min-cut partitioning issue is the partitioning determined using Algorithm 2. (Definition 5).

Rooted trees must adhere to a clan limitation. To until point, we have been concerned mostly with uprooted trees. These options reasoned in part by the fact that most phylogenetic reconstruction techniques use timereversible theories of sequence evolution (such GTR [19]), which produce an unrooted tree. However, researchers have devised a number of approaches for rooting trees [20, 21], such as accurate and linear-time methods as MV rooting [16]. Each "monophyletic clade," or collection of things that comprises all offspring of their common ancestor, is a biologically relevant unit when a rooted tree is available. Because of this, it's possible that we'd want to force every grouping to be a clade. This "clade" restriction simplifies clustering since our methods can be simply modified to check whether each cluster is also a clade. To be more precise, both (u, ul) and (u, ur) need to be cut (instead of just the longer one) when B(ul) + wl + B(ur) + wr >, as in Algorithm 1. To solve the Max-diameter, Sum-length, and Single-linkage min-cut partitioning issues under the clade constraint in linear time, we make a little tweak to the original algorithm.

Sequence centered on a central point. Most sequencing-based clustering techniques provide a representative sequence for each cluster, which is then often re-used by the algorithm itself. We don't use representatives in our clustering method. On the other hand, if a representative is required for downstream.

When it comes to software, you have options. The node that minimizes the variation of root-to-tip lengths, or the cluster's midpoint, may be located in linear time [16], and the representative leaf can then be chosen based on its proximity to the midpoint.

The consensus sequence among all cluster sequences is another option (i.e., choosing the most frequent letter for each site). In certain cases, a consensus sequence might be used instead of one of the provided sequences as the central sequence [22]. Ancestral sequence reconstruction is a third possibility that we investigate in our findings. We begin by establishing a balancing point as the first node of each subtree specified by the cluster. The reconstructed root sequence is then used as the center of our maximum likelihood ancestral state reconstruction (ASR) analysis.

Computer program called TreeCluster

In a publicly accessible open source program called TreeCluster, we developed linear-time algorithms for the min-cut partitioning issue under Maxdiameter, Sum-branch, Single-linkage, and other clustering criteria, with and without clade restrictions. Inputs to TreeCluster include a newick tree and a threshold value; the program then outputs clusters in a text file. TreeCluster relies on the treeswift [23] package to perform tree operations quickly.

Tree Cluster's three uses

Although sequence clustering has various uses, in this study we focus on three of them.

First use case: grouping OTUs. The issue is biological. Pipeline standards use operational

taxonomic units for microbiome studies based on 16S sequences derived from community-wide samples (OTUs). The OTUs are the most fine-grained degree of differentiation between species, and they are created by clustering sequences that have at least some amount of similarity (e.g., 97% similarity).

When doing further studies like taxonomic profiling, taxonomic identification, sample differentiation, or machine learning, all sequences assigned to the same OTU are considered to belong to the same organism. Similarity thresholds are increasingly being used in place of biological species concepts.

Additionally, OTUs comprised of clusters of similar sequences might provide some resistance against sequencing mistakes.

Most uses of operational taxonomic units (OTUs) are "closed-reference," meaning that OTUs are created for reference sequences using techniques like UCLUST [2] and Dotur [3]. These techniques group sequences together based on a specified similarity criterion, often selecting a centroid sequence to stand in for an operational taxonomic unit (OTU). Sequence similarity is used to determine the relative distance between individual reads from a 16S sample and the OTUs, and the closest OTU is determined for each read. When all of the reads have been processed for all of the samples, an OTU table may be constructed in which the rows stand for the samples, the columns stand for the OTUs, and the cells stand for the frequency of the OTU in the sample. After this table is established, more research may be conducted. Such OTU-based studies may make use of one of many sizable reference datasets already available [26-28]. Among them is Greengenes [28], which has gained traction because to pipelines like Qiita [29].

In order to provide the most accurate representation of organisms, an OTU table should include OTUs that are as coherent as feasible (i.e., internally consistent). For our investigations, we will use Greengenes as our reference library and concentrate on closed-reference OTU selecting techniques. It is important to keep in mind, however, that there are other approaches that require the grouping of sequences for the same purpose, such as open-reference OTU picking and sub-operational taxonomic unit (sOTU) methods [30-32].

techniques already in use. Non-hierarchical clustering approaches [2, 34] are more popular than hierarchical ones [3, 33] for OTU clustering, perhaps because of the reduced processing load. For a given threshold, UCLUST dynamically determines a set of representative sequences by assigning query sequences into representative sequences (centroids) such that, ideally, the distance between each query and its assigned centroid is less than while distances between centroids is more than. CD-HIT [34] is another well-known method that uses a similar algorithmic strategy. UCLUST is a heuristic method, therefore the grouping it produces might be different depending on the sequence in which the queries were processed. CD-HIT is distinct from UCLUST principally because to its method of distance calculation.

Partioning as a minimum-cut formulation. Using an estimated ML phylogeny and a threshold, we create OTUs by solving the Min-diameter, Sum-Length, or Single-linkage min-cut partitioning issues. To identify OTUs, we label each cluster in the resultant partition.

Experiments. Tree-based OTU clustering is compared to UCLUST, which is employed by Greengenes [28] to determine its efficacy. Here, we apply TreeCluster to the phylogenetic tree of 203,452 sequences from the Greengenes v13.5 database using three different modes: max, sum, and single-linkage. Twenty criteria are used, including [0.005, 0.05] with a 0.005-step size and [0.05, 0.15] with a 0.01-0.01-step size. Because the number of clusters becomes substantially less over the 0.1 threshold compared to other approaches, we limit our exploration of single-linkage to that value.

Clusters of operational taxonomic units (OTUs) are retrieved from the Greengenes database at sequence identity cutoffs ranging from 0.1 to 0.15. (i.e., 0.03, 0.06, 0.09, 0.12, and 0.15). For each clustering L1,..., LN,' we calculate the weighted average of average pairwise distance per cluster (which we term cluster diversity for brevity) using the method below.

International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021

$$\mu(\{L_1,\ldots,L_N\}) = \frac{\sum_{k=1}^{N} |L_k| \frac{\sum_{i,j \in L_k} d(i,j)}{|L_k|^2}}{\sum_{k=1}^{N} |L_k|} = \frac{1}{n} \sum_{k=1}^{N} \sum_{i \in L_k} \frac{d(i,j)}{|L_k|} \quad (1)$$

in where n is the total number of sequences that have been grouped together. We use two approaches to calculate the distance d(i, j) between two elements: route length on the inferred

sequence-based Hamming distance and phylogenetic tree. All 203,452 sequences in the Greengenes database are aligned in a multiple sequence alignment, and the Hamming distances are calculated pairwise from this alignment, ignoring any site that has a gap in the pairwise alignment. Distinctly, a cluster

Diversity is necessary but not sufficient for evaluating outcomes (singletons have zero diversity). Instead, we evaluate the variety of approaches at the same clustering depth. Consequently, we evaluate approaches for selections of the threshold where they produce (nearly) similar numbers of clusters as we vary. If two methods produce the same number of clusters, the one with less variability among those clusters is preferred.

We use two criteria to assess the quality of a sample sequence set. Let's say we have a clustering L1,..., LN, and we want to designate all of the non-singleton clusters as L1,..., LN0. One measure is the average distance from each cluster's center, or centroid, as defined technically as:

$$v(g,\{L_1,\ldots,L_N\}) = \frac{1}{N} \sum_{k=1}^{N} \sum_{i \in L_k} \frac{d(i,g(L_k))}{|L_k|} \quad (2)$$

in where g is a function that converts a cluster into a (representative) sequence. Maximum average distance to cluster representatives is the second measure, and it is technically defined as:

$$\zeta(g,\{L_1,\ldots,L_N\}) = \frac{1}{N} \sum_{k=1}^{N} \max_{i \in L_k} d(i,g(L_k)) \quad (3)$$

Since a naive clustering that assigns many singletons would have a trivially low value for v and, we define these metrics on the set of non-singleton clusters (near zero).

The Greengenes database comes with a sample set of sequences that have already been calculated for you. There are a several different ways to choose a centroid for a TreeCluster, but we'll be looking at consensus and ASR. We use TreeTime [35] with the GTR model to conduct ASR. To estimate GTR model parameters, we align sample sequences from Greengenes using a multiple sequence alignment with a 15 percent identity threshold and run the alignment using RAxML 8 [36]. The Hamming distance between two items is calculated as d(i, j).

Second Use: Analyses of HIV transmission clusters. The issue is biological. Due of HIV's high rate of evolution, phylogenetic connections between sequences may reveal its transmission history [37]. In order to effectively plan and assess HIV control measures, epidemiologists rely heavily on the results of phylogenetic studies of HIV sequences [38-42]. The data from these studies may provide light on questions of genetic relatedness [43], transmission histories [44], and inter-population mixing [45]. Infer transmission clusters from pairwise sequence distances, track the growth of clusters over time, and prioritize clusters with the highest growth rates [46] are all examples of how transmission clustering is being used to predict at-risk individuals and the spread of epidemics using computational molecular epidemiology. As a result of this monitoring paradigm, two problems arise: (1) How may transmission clusters be inferred most effectively from molecular data, and (2) What is the best technique to infer transmission clusters?

techniques already in use. We zero in on two widely-used programs that conduct this kind of grouping. The sequences, a phylogenetic tree, and a distance criterion are input into Cluster Picker [4]. It groups people into clusters where the largest pairwise sequence-based distance is below the threshold, the number of clusters is reduced, and each cluster determines the leaves of a clade in the tree. By comparing the Tamura-Nei 93 (TN93) distance [47] between two people u and v, HIV-TRACE may determine whether or not they belong to the same cluster [5]. Each of these approaches

scales poorly with the number of sequences (quadratic and cubic, respectively).

respectively (hours or days for HIV-TRACE and Cluster Picker to execute on huge datasets) (however, HIV-TRACE enjoys trivial parallelism and is fast in practice).

Partioning as a minimum-cut formulation. Clustering in transmission is analogous to our own concept in that it includes cutting edges to produce clusters that must meet specific requirements (as specified by the leafsets produced by the cuts). In its original form, Cluster Picker is similar to our Max-diameter min-cut partitioning (with the additional constraint that clusters must define clades in the phylogeny), and HIV-TRACE is similar to our Single-linkage min-cut partitioning. Both methods use pairwise distances computed from sequences.

Experiments. We initially use FAVITES [48] to simulate HIV epidemic data in order to assess the efficacy of clustering in preventing the spread of the virus. To mimic the HIV pandemic in San Diego from 2005 to 2014, we use the simulation parameters reported by Moshiri et al. [48]. While the initial parameter set had all HIV patients sequenced after the conclusion of the pandemic, producing an ultrametric tree in the unit of time, we changed this to sequence all patients at their first initiation of antiretroviral therapy (ART). We run simulations with two different parameters: when ART is assumed to be initiated and how extensive the underlying social interaction network is thought to be. There is a correlation between higher ART rates and lower degree requirements, with the former causing a delayed pandemic and the latter altering patterns of evolutionary branch length [48]. To access the full FAVITES parameter set, please refer to the appendices (List A in S1 Appendix).

FastTree-II [8] is used to infer phylogenies from synthetic sequences under the GTR+ model, and then FastRoot [49] is used to root the trees using the MinVar approach.

We infer clusters of transmission using HIV-TRACE [5] and several clustering strategies in TreeCluster. Cluster Picker [4] took too long to execute, thus we couldn't utilize it.

According to HIV-authors TRACE's [46], a clustering threshold of 1.5% is used.

We employ a clustering threshold of 3% for Single-Linkage TreeCluster because HIV-TRACE predicts pairwise sequencing distances under the TN93 model, [47] which likely to be underestimates of phylogenetic distance computed under the GTR model. Since the Max-diameter clustering criterion in Cluster Picker is predefined at 4.5 percent [4], we employ this value for Max-Diameter TreeCluster (both with and without the Clade constraint). We just double the Max-diameter threshold and set it to 9% for usage in Sum-length TreeCluster (with and without the Clade constraint). We use these criteria as a starting point, but we also evaluate a broad variety of thresholds for each transmission clustering approach to ensure they are resilient.

In order to identify the 1,000 most important people, we first assess cluster growth from year 8 to year 9 of the simulation, and then rank them in decreasing order of respective growth. The risk associated with a certain person, u, is quantified by tallying the number of u!v HIV transmission episodes that occurred between the ages of 9 and 10. We use the mean risk of the top one thousand people as a proxy for the efficacy of a particular clustering. It's preferable to aim for more than 1,000 people since it means more transmissions will be stopped. We also provide the predicted risk, or the average number of transmissions throughout a population, based on a sample size of 1,000 people chosen at random.

Thirdly, we apply this technique to multiple sequence alignment by breaking it down into manageable chunks. Consider an example of an algorithm.

Some researchers have even tried employing tree-based clustering as a kind of divide-and-conquer for multiple sequence alignment (MSA). The tree structure may be used to partition sequences into subsets (i.e., clusters) that can be aligned independently and then combined to solve the MSA issue using the divide-and-conquer strategy. Iterating back and forth between tree and MSA inference allows for simultaneous inference of both the phylogeny and the MSA, and has been implemented in methods like as

PASTA [52] and SATe [50, 51]. The divide-and-conquer strategy has been shown to be especially helpful for multi-stage aggregation (MSA) of extremely large datasets [9, 10, 50]. Not all MSA tools employ divide-and-conquer, and the use of min-cut partitioning in divide-and-conquer is the only thing we look at in this paper methods. Using PASTA [52], a scalable program that infers both MSAs and trees for ultra-large datasets (tested for up to 1,000,000 sequences), we analyze the efficacy of min-cut partitioning.

Before clustering the sequences, PASTA constructs a rough phylogenetic estimate. PASTA's "divide" stage involves clustering the input sequences into subgroups, with the goal of reducing the diversity of each subset relative to the overall set. The MSA and/or tree are then inferred using a precise (but often computationally intensive) approach applied to the subsets.

At last, a number of methods are used to combine the findings from the individual subgroups. Both the efficacy of the technique used to partition the tree into subsets and the accuracy of the base method used on the subsets and merging method contribute to the accuracy of the output [51].

PASTA first generates an alignment with the help of HMMs implemented in HMMER [53] and a tree with the help of FastTree-II [8]; it then performs several iterations (3 by default) of the divide-and-conquer strategy described, using MAFFT [54] for aligning subsets and a combination of OPAL [55] and a technique based on transitivity for merging subalignments.

At the conclusion of each iteration, FastTree-II produces a tree that serves as the starting point for the subsequent iteration. The approach has been proved to be very accurate on both synthetic and actual data, notably in terms of tree correctness, where it is almost on par with the accuracy achieved using the genuine alignment. It has much potential for development, however, in terms of alignment accuracy, especially on the most difficult datasets.

Centroid-edge decomposition is the foundation of PASTA's clustering algorithm. The decomposition is defined recursively, starting with the guide tree from the previous iteration, and dividing the tree into two pieces of equal size (or as near to equal size as feasible). Once that's done, recurse on each subtree until there are no more than 200 leaves remaining.

Partioning as a minimum-cut formulation. The edges must be cut and a constraint must be defined on the subsets for the centroid edge decomposition to be valid. However, it lacks optimization of any inherent objective function due to its procedurally created nature. The restrictions of the min-cut partitioning are similar to those of the centroid decomposition, but the result is different.

Here, we reduce the number of subsets by solving the Sum-length min-cut partitioning problem with a threshold of = 2m 2. This is accomplished by setting all of the edge weights of the guide tree to 1. As a result, the requirements of this "max-size min-cut partitioning" are the same as those of centroid decomposition, but it ensures finding the fewest possible clusters.

Experiments. We run PASTA version 1.8.3 on two datasets, and for each, we compare the accuracy of the two decomposition algorithms, to see how our novel decomposition affects PASTA. Partitioning based on centroid and maximum size minimum cut. Both decomposition methods maintain the same maximum subset size and other characteristics. To test our models, we utilized data from the original PASTA article, namely, 19 different real-world HomFam datasets containing anything from 10,099 to 93,681 protein sequences, and 10 replication of a simulated RNAsim dataset including 10,000 leaves. The RNASim was developed using a very intricate model of RNA evolution. Here, we make use of the simulation-determined true alignment as a standard. Since the real alignment for HomFam is unknown, we adopt the approach of using a limited set of seed sequences that have been manually selected to have a solid alignment as a reference [9, 56]. Alignment errors are quantified in both scenarios by means of two industry-standard measures calculated using FastSP [57]: Total SPFN (the share of

inconsistencies between the calculated alignment and the reference alignment (i.e., missing homologies) and SPFP (the percentage of homologies in the estimated alignment not present in the reference).

Results

International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021

Clustering of OTUs (First Application) Results

By adjusting the threshold from 0.005 to 0.15, we get 181, 574 to 10, 112 clusters in the Greengenes dataset (note that singletons are also counted). The cluster diversity decreases in a non-linear fashion with increasing cluster formation thresholds (Fig 2A and S1 Fig). When evaluating the three goals,

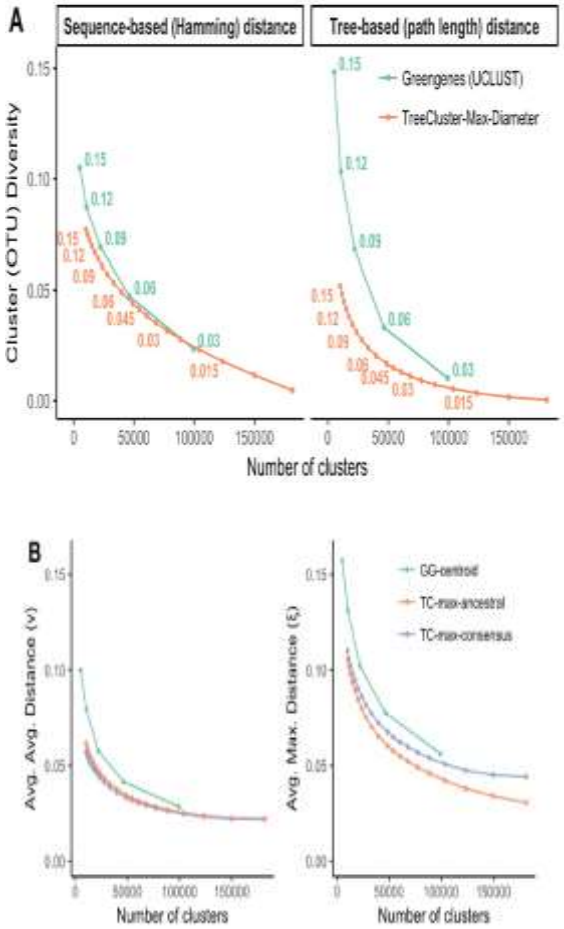


Diagram of the similarities and differences between Greengenes and TreeCluster. Greengenes and TreeCluster (A) cluster diversity (Eq 1) vs OTU count. When evaluating cluster diversity, we use both hamming distance and tree-based distance. For all Greengenes data points and for certain TreeCluster data points, the threshold is shown. Check out S1 Fig to see how the various TreeCluster modes stack up against one another.

Greengenes and TreeCluster's average-average (v) and average-maximum () distance to the centroid vs cluster count (B). The centers of TreeClusters may be calculated either by consensus or by reconstructing ancestral states.

Max-diameter and Sum-length show comparable tendencies of cluster diversity scores when compared to other TreeCluster functions, however Single-linkage min-cut partitioning has much more diversity than the other two (S1 Fig). In this pattern, seen whether using tree distances or sequence distances, with differences being greater for tree distances. Finally, it is important to note that when calculated using tree distances, cluster diversity is smaller, indicating that clusters are close in phylogenetic space, despite the fact that tree distances are, as predicted, bigger than sequence distances (S2 Fig).

Maxdiameter min-cut partitioning creates more compact clusters for tree-based scores than the standard Greengenes OTUs created using UCLUST (Fig 2A). Cluster diversity scores for Greengenes OTUs are much lower when using tree distance to assess sequence distances, and the disparity widens for higher thresholds. By way of illustration, for = 0.15, the cluster diversity of Greengenes OTUs is three times that of TreeCluster OTUs. Greengenes and TreeCluster both produce about the same number of clusters at low threshold values (e.g., = 0.03 for Greengenes is comparable to = 0.02 for TreeCluster) when comparing sequences using the Hamming distance.

Remarkably, nevertheless, TreeCluster beats Greengenes OTUs by as much as 1.4-fold (e.g., = 0.15) when the number of OTUs is drastically decreased. Despite the fact that UCLUST uses sequence distances, TreeCluster does not, this is the case.

Greengenes's biggest cluster is much bigger than TreeCluster's (Table 1). For = 0.09, for instance, the number of clusters produced by Greengenes and TreeCluster is quite close (22.090 and 23.631 clusters, respectively), while the greatest cluster size in Greengenes is 3.0 times that in TreeCluster (1,659 versus 540). However, at the same threshold value, 48% of Greengenes clusters are singletons but only 27% of TreeCluster clusters are singletons. Therefore, GreenGenes has a larger proportion of extreme cluster sizes compared to TreeCluster.

Representative sequences in TreeCluster are more closely related to other cluster sequences than Greengenes, whether the distance is calculated using a consensus or an ASR technique (Fig 2B).

International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021

The v-score reveals a little performance gap between using consensus centroids and ASR representative sequences (e.g., = 0.062 and = 0.057, respectively, when = 0.15). ASR representative sequences perform somewhat better than consensus at all threshold levels (e.g., = 0.03 and = 0.04 respectively when = 0.005), and the difference again rises as the number of clusters increases. TreeCluster computes better centroids for both sorts of sequences (e.g., up to 1.7-fold when = 0.15 for) than Greengenes representative sequences do for either measure.

Application 2 Outcomes Dynamics of HIV

Regardless of the settings we change, when we compare different TreeCluster modes, When compared to other clustering algorithms, Sum-length TreeCluster always performs better, and adding the Clade requirement has little influence on performance (Fig 3). The average risk of the top 1,000 people from Sum-length clusters is 0.85, which is much higher than the estimated risk of 0.55 transmissions for a random selection of individuals. Maximum diameter in TreeClusters is a close second to sum length in all cases. TreeCluster's two modes are much more efficient than most alternatives.

Single-linkage TreeCluster and HIV-TRACE consistently perform worse than the other methods when varying the expected time to begin ART and expected degree, with Single-linkage TreeCluster typically performing around the theoretical expectation of a random selection and HIV-TRACE performing slightly better (Fig 3a and 3b). In addition, these


PLOS ONE

TreeCluster: Clustering biological sequences using phylogenetic trees

Table 1. Number of singleton clusters (s), total number of clusters (T), and maximum cluster size (max) for TreeCluster and GreenGenes for various thresholds. In the Greengenes database, OTU definitions for thresholds = 0.015 and = 0.045 are not available.

	TreeCluster Max-Diameter			GreenGenes OTU 13_5		
	s	T	max	s	T	max
0.015	86387	115836	47	(n.a.)	(n.a.)	(n.a.)
0.03	43510	57269	96	70465	99522	527
0.045	24795	34088	171	(n.a.)	(n.a.)	(n.a.)
0.06	15257	19809	305	26485	46256	894
0.09	6396	25431	540	10560	22090	1659
0.12	3005	13052	800	4353	10544	2331
0.15	1525	10112	1299	1735	5088	3745

https://doi.org/10.1371/journal.pone.0221068.t001

Single-linkage TreeCluster and HIV-TRACE regularly perform worse than predicted by random selection, demonstrating trends that are not just attributable to the selected criteria (Fig 3c). Importance of Sum-length

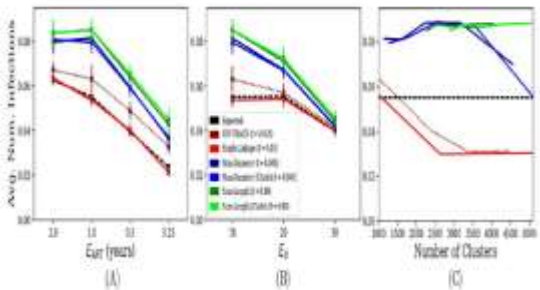
TreeCluster's Max-diameter setting is optimal for datasets with between 2,000 and 5,000 nodes, whereas the same setting is optimal for datasets with between 2,000 and 3,000 nodes.

The Outcome of the Third Application: Enhancing PASTA

To begin, we find that PASTA 1.8.3 is a significant upgrade over the 2015 version [52] released version, notably in terms of alignment accuracy for RNASim (by roughly 3%). This occurred because significant improvements were made to the PASTA program and associated tools. The alignment error is much reduced for the RNASim dataset when Max-size min-cut partitioning is used in PASTA, but only somewhat for the HomFam dataset (Fig. 4). The average SPFN on RNAsim data decreases from 0.12 to 0.10, which is a 17% improvement in accuracy. These reductions are robust between replicates and noteworthy since the only modification we made to PASTA was to replace its deconstruction phase with our new clustering technique. In particular, the strategies for aligning and merging subsets and for inferring trees remained unchanged. While errors were reduced in the HomFam dataset, they were not drastically reduced (Fig 4b). Given these findings, we have updated PASTA to use Max-size min-cut partitioning by default.

Discussion

Several theoretical and practical issues should be further discussed.



International Journal of Information Technology & Computer Engineering

SNNO: 2347–3657

Volume 9,Issue3 July ,2021

Transmission clustering's efficiency (Figure 3). When evaluating the success of a vaccination campaign, the average number of people infected by the chosen 1,000 is used as the effectiveness metric. Predicted ART initiation time (A), predicted contact network degree (B), and cluster size (C) are shown along the horizontal axis.

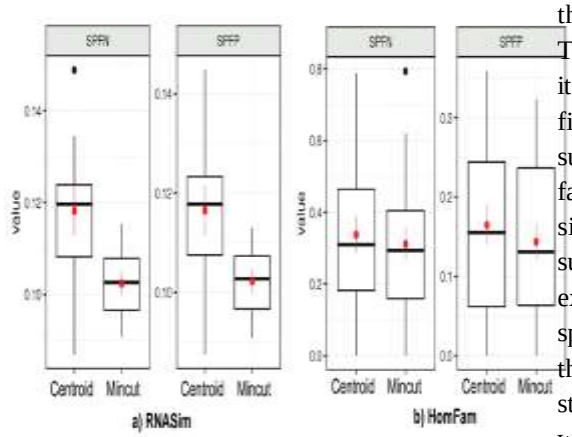


Fig 4. Alignment error for PASTA using the centroid and the mincut decompositions. We show Sum of Pairs False Negative (SPFN) and Sum of Pairs False Positive (SPFP) computed using FastSP [57] over two datasets: the simulated RNASim dataset (20 replicates) and the biological HomFam dataset (19 largest families; all 20 largest, except "rbr" omitted due to the warning on the Pfam website). We show boxplots

Mean-diameter min-cut partitioning

Cluster Picker [4] is one of the available approaches that uses mean pairwise distance between nodes to establish its constraints. One may define a variant of the mincut partitioning issue that is similar to these by saying that

$$f_T(L) = \frac{1}{\binom{|L|}{2}} \sum d(i, j).$$

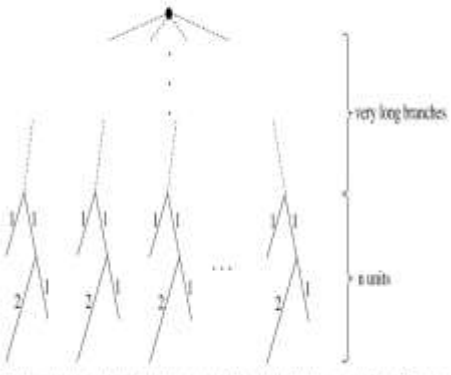
Unfortunately, our greedy approach can only solve the "Mean-diameter" min-cut partitioning issue in linear time if we additionally include clade restrictions (Algorithm B in S1 Appendix). In light of the contrasting case,

The lack of clade restrictions in S3 Fig causes the greedy method to fail. Using mean as a function fT ( ) adds another layer of complexity, and whether or not it can be solved in linear time is still debatable. It's unclear whether mean diameter is a valid metric. For instance, it is feasible that a cluster's mean diameter is below the threshold, but the mean diameter of subclusters nested inside that cluster is not; such circumstances may not make sense for downstream applications.

Collection of Best Answers

There may be numerous optimum solutions to each of our min-cut partitioning issues, each using a different partition with the same number of clusters. Furthermore,

As can be shown in Fig. 5, the number of possible optimum solutions might grow exponentially with the number of nodes in a binary phylogenetic tree. The sheer number of optimum solutions may make it impracticable to list them all, which is what this finding suggests. Finding a mechanism to summarize all optimum partitions, however, is still fascinating and might be useful in certain situations. Unfortunately, such a method of summarizing is not yet available to us. Despite the exponentially vast size of the optimum solution space, the set of all edges that might exist in any of the ideal solutions can be readily determined, as stated in Lemma A of S1 Appendix. As a result, we were able to locate unbreakable edges that would not be severed by any data clustering algorithm.



Optional criteria selection

Max-diameter and Sum-diameter, two of the three approaches we considered, were shown to consistently outperform Single-linkage. This finding seems reasonable.

Due to the transitive nature of its criteria, single-linkage may boost variety inside a cluster. As a result, transitivity may allow for the collapse of a

highly diverse dataset into a single cluster. Our interest in finding a solution to the Single-linkage issue was sparked by the fact that HIV-TRACE, the most popular HIV clustering approach, employs a similar idea. We found no benefit to using this method of clustering over Max-diameter or Sum-length, thus we suggest those two alternatives instead.

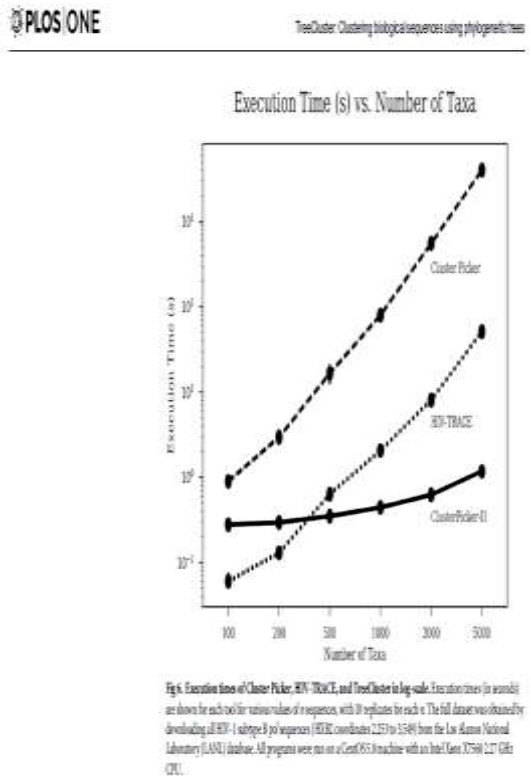
Max-diameter has the upper hand since its threshold is more intuitive.

Finally, we highlight that although calculating consensus sequences is considerably simpler and quicker, ASR-based selection of representative sequences fared better.

The current elapsed time

Our primary comparison was TreeCluster's efficacy, but we also found that its running time (after the tree is inferred) was competitive with that of other clustering algorithms. We used HIV-TRACE, Cluster Picker, and TreeCluster to a real-world HIV dataset, processing subsets of data with 100-5,000 sequences (Fig 6). TreeCluster's processing time on the biggest set with 5,000 leaves did not surpass 2 seconds. HIV-Trace, which relies on sequences, took around a minute, which is very quick, whereas Cluster Picker took almost an hour. TreeCluster was able to accomplish clustering in under 30 seconds even on the Greengenes dataset, which has more than 200,000 leaves. As a result of TreeCluster's speed, we can rapidly test how different criteria affect the results of our downstream applications.

These estimates do not take into account the time required to infer the tree if one is not already available (although in many cases, a tree is inferred for other reasons and is thus already accessible). Just as an example, you may look at the data from



For example, using PASTA and 12 CPUs, MSA and tree inference on datasets containing 10,000 sequences might take close to an hour. It is estimated that around a third of this time is spent on tree inference (for example, see Fig 4 of [9]), while the other two-thirds is spent on the estimating alignment, which is also required by most alternative methods.

Conclusion

We presented TreeCluster, a technique that uses several optimization objective functions to cluster sequences at the branch tips of a phylogenetic tree. Several downstream applications, such as OTU clustering, HIV transmission clustering, and divide-and-conquer alignment, were shown to make advantage of our linear-time techniques. The clusters' internal coherence and the quality of subsequent analysis are both enhanced by using the tree to construct them.

References

1. Goodrich JK, Di Rienzi SC, Poole AC, Koren O, Walters WA, Caporaso JG, et al. Conducting a microbiomestudy. Cell. 2014; 158(2):250–262. https://doi.org/10.1016/j.cell.2014.06.037 PMID: 25036628

2. Edgar RC. Search and clustering orders of magnitude faster than BLAST. Bioinformatics. 2010; 26(19):2460–2461. https://doi.org/10.1093/bioinformatics/btq461 PMID: 20709691

3. Schloss PD, Handelsman J. Introducing DOTUR, a Computer Program for Defining Operational TaxonomicUnits and Estimating Species Richness. Applied and Environmental Microbiology. 2005; 71(3):1501–1506. https://doi.org/10.1128/AEM.71.3.1501-1506.2005 PMID: 15746353

4. Ragonnet-Cronin M, Hodcroft E, Hue´ S, Fearnhill E, Delpech VC, Brown AJL, et al. Automated analysisof phylogenetic clusters. BMC bioinformatics. 2013; 14:317. https://doi.org/10.1186/1471-2105-14-317PMID: 24191891

5. Kosakovsky Pond SL, Weaver S, Leigh Brown AJ, Wertheim JO. HIV-TRACE (TRAnsmission ClusterEngine): a Tool for Large Scale Molecular Epidemiology of HIV-1 and Other Rapidly Evolving Pathogens.Molecular Biology and Evolution. 2018; 35(7):1812–1819. https://doi.org/10.1093/molbev/msy016 PMID: 29401317

6. Hillis DM, Bull JJ. An Empirical Test of Bootstrapping as a Method for Assessing Confidence in PhylogeneticAnalysis. Systematic Biology. 1993; 42(2):182–192. https://doi.org/10.2307/2992540

7. Warnow T. Computational phylogenetics: An introduction to designing methods for phylogeny estimation.Cambridge University Press; 2017.

8. Price MN, Dehal PS, Arkin AP. FastTree 2—Approximately maximum-likelihood trees for large alignments.PLoS ONE. 2010; 5(3). https://doi.org/10.1371/journal.pone.0009490

9. Mirarab S, Nguyen N, Guo S, Wang LS, Kim J, Warnow T. PASTA: Ultra-Large Multiple SequenceAlignment for Nucleotide and Amino-Acid Sequences. Journal of Computational Biology. 2015; 22(05):377–386. https://doi.org/10.1089/cmb.2014.0156 PMID: 25549288

10. Nguyen NPDNpD, Mirarab S, Kumar K, Warnow T. Ultra-large alignments using phylogeny-aware profiles.Genome Biology. 2015; 16(1):124. https://doi.org/10.1186/s13059-015-0688-z PMID: 26076734

11. Enright AJ, Van Dongen S, Ouzounis CA. An efficient algorithm for large-scale detection of protein families.Nucleic acids research. 2002; 30(7):1575–84. https://doi.org/10.1093/nar/30.7.1575 PMID: 11917018

12. Li L, Stoeckert CJ, Roos DS. OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes.Genome Research. 2003; 13(9):2178–2189. https://doi.org/10.1101/gr.1224503 PMID: 12952885

13. Steinegger M, So¨ding J. MMseqs2 enables sensitive protein sequence searching for the analysis ofmassive data sets. Nature Biotechnology. 2017. https://doi.org/10.1038/nbt.3988 PMID: 29035372

14. Nguyen NP, Warnow T, Pop M, White B. A perspective on 16S rRNA operational taxonomic unit clusteringusing sequence similarity. npj Biofilms and Microbiomes. 2016; 2:16004. https://doi.org/10.1038/npjbiofilms.2016.4 PMID: 28721243

15. Ragonnet-Cronin M, Hodcroft E, Hue´ S, Fearnhill E, Delpech V, Brown AJL, et al. Automated analysisof phylogenetic clusters. BMC bioinformatics. 2013; 14(1):317. https://doi.org/10.1186/1471-2105-14-317 PMID: 24191891

16. Mai U, Sayyari E, Mirarab S. Minimum variance rooting of phylogenetic trees and implications for speciestree reconstruction. PLOS ONE. 2017; 12(8):e0182238. https://doi.org/10.1371/journal.pone.0182238 PMID: 28800608

17. Parley A, Hedetniemi S, Proskurowski A. Partitioning trees: Matching, domination, and maximum diameter.International Journal of Computer & Information Sciences. 1981; 10(1):55–61. https://doi.org/10.1007/BF00978378

18. Kundu S, Misra J. A Linear Tree Partitioning Algorithm. SIAM Journal on Computing. 1977; 6(1):151–154. https://doi.org/10.1137/0206012

19. Tavare´ S. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. Lectures onMathematics in the Life Sciences. 1986; 17:57–86.

20. Kluge AG, Farris JS. Quantitative Phyletics and the Evolution of Anurans. Systematic Biology. 1969; 18(1):1–32. https://doi.org/10.1093/sysbio/18.1.1

21. Farris JS. Estimating Phylogenetic Trees from Distance Matrices. The American Naturalist. 1972; 106(951):645–668. https://doi.org/10.1086/282802

22. Zheng Q, Bartow-McKenney C, Meisel JS, Grice EA. HmmUFOtu: An HMM and phylogenetic placementbased ultra-fast taxonomic assignment and OTU picking tool for microbiome amplicon sequencingstudies. Genome Biology. 2018; 19(1):82. https://doi.org/10.1186/s13059-018-1450-0 PMID: 29950165

23. Moshiri N. TreeSwift: a massively scalable Python tree package. bioRxiv. 2018. https://doi.org/10.1101/325522

24. Schloss PD, Westcott SL. Assessing and Improving Methods Used in Operational Taxonomic Unit-Based Approaches for 16S rRNA Gene Sequence Analysis. Applied and Environmental Microbiology.2011; 77(10):3219–3226. https://doi.org/10.1128/AEM.02810-10 PMID: 21421784

25. Chen W, Zhang CK, Cheng Y, Zhang S, Zhao H. A Comparison of Methods for Clustering 16S rRNASequences into OTUs. PLoS ONE. 2013; 8(8):e70837. https://doi.org/10.1371/journal.pone.0070837PMID: 23967117

26. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, et al. The SILVA ribosomal RNA genedatabase project: improved data processing and web-based tools. Nucleic acids research. 2012; p.gks1219. https://doi.org/10.1093/nar/gks1219

27. Maidak BL. The RDP-II (Ribosomal Database Project). Nucleic Acids Research. 2001; 29(1):173–174.https://doi.org/10.1093/nar/29.1.173 PMID: 11125082

28. DeSantis TZ, Hugenholtz P, Larsen N, Rojas M, Brodie EL, Keller K, et al. Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. Appl Environ Microbiol. 2006; 72(7):5069–5072. https://doi.org/10.1128/AEM.03006-05 PMID: 16820507

29. Gonzalez A, Navas-Molina JA, Kosciolek T, McDonald D, Va´zquez-Baeza Y, Ackermann G, et al. Qiita:rapid, web-enabled microbiome meta-analysis. Nature Methods. 2018; 15(10):796–798. https://doi.org/10.1038/s41592-018-0141-9 PMID: 30275573

30. Amir A, McDonald D, Navas-Molina JA, Kopylova E, Morton JT, Zech Xu Z, et al. Deblur RapidlyResolves Single-Nucleotide Community Sequence Patterns. mSystems. 2017; 2(2). https://doi.org/10.1128/mSystems.00191-16

31. Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. DADA2: High-resolutionsample inference from Illumina amplicon data. Nature Methods. 2016; 13:581. https://doi.org/10.1038/nmeth.3869 PMID: 27214047

32. Edgar RC. UNOISE2: improved error-correction for Illumina 16S and ITS amplicon sequencing. bioRxiv.2016. https://doi.org/10.1101/081257

33. Cai Y, Sun Y. ESPRIT-Tree: hierarchical clustering analysis of millions of 16S rRNA pyrosequences inquasilinear computational time. Nucleic Acids Research. 2011; 39(14):e95–e95. https://doi.org/10.1093/nar/gkr349 PMID: 21596775

34. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotidesequences. Bioinformatics. 2006; 22(13):1658–1659. https://doi.org/10.1093/bioinformatics/btl158PMID: 16731699

35. Sagulenko P, Puller V, Neher RA. TreeTime: Maximum-likelihood phylodynamic analysis. Virus

Evolution.2018; 4(1):1–9. https://doi.org/10.1093/ve/vex042

36. Stamatakis A. RAxML version 8: A tool for phylogenetic analysis and post-analysis of large phylogenies.Bioinformatics. 2014; 30(9):1312–1313. ttps://doi.org/10.1093/bioinformatics/btu033 PMID:24451623

37. Leitner T, Escanilla D, Franzen C, Uhlen M, Albert J. Accurate reconstruction of a known HIV-1 transmissionhistory by phylogenetic tree analysis. Proceedings of the National Academy of Sciences. 1996;93(20):10864–10869. https://doi.org/10.1073/pnas.93.20.10864

38. Aldous JL, Pond SK, Poon A, Jain S, Qin H, Kahn JS, et al. Characterizing HIV transmission networksacross the United States. Clinical infectious diseases: an official publication of the Infectious DiseasesSociety of America. 2012; 55(8):1135–43. https://doi.org/10.1093/cid/cis612

39. Hue´ S, Clewley JP, Cane PA, Pillay D. HIV-1 pol gene variation is sufficient for reconstruction of transmissions in the era of antiretroviral therapy. AIDS (London, England). 2004; 18(5):719–28. https://doi. org/10.1097/00002030-200403260-00002

40. Hughes GJ, Fearnhill E, Dunn D, Lycett SJ, Rambaut A, Leigh Brown AJ, et al. Molecular phylodynamics of the heterosexual HIV epidemic in the United Kingdom. PLoS pathogens. 2009; 5(9): e1000590. https://doi.org/10.1371/journal.ppat.1000590 PMID: 19779560

41. Leigh Brown AJ, Lycett S, Weinert L, Hughes G, Fearnhill E, Dunn DT. Transmission network parameters estimated from HIV sequences for a nationwide epidemic. Journal of Infectious Diseases. 2011; 204(9):1463–1469. https://doi.org/10.1093/infdis/jir550 PMID: 21921202

42. Mehta SR, Kosakovsky Pond SL, Young JA, Richman D, Little S, Smith DM. Associations between phylogenetic clustering and HLA profile among HIV-infected individuals in San Diego, California. Journal of Infectious Diseases. 2012; 205(10):1529–1533. https://doi.org/10.1093/infdis/jis231 PMID: 22448013

43. Eshleman SH, Hudelson SE, Redd AD, Wang L, Debes R, Chen YQ, et al. Analysis of genetic linkage of HIV from couples enrolled in the HIV prevention trials network 052 trial. Journal of Infectious Diseases. 2011; 204(12):1918–1926. https://doi.org/10.1093/infdis/jir651 PMID: 21990420

44. Hue´ S, Brown AE, Ragonnet-Cronin M, Lycett S, Dunn DT, Fearnhill E, et al. Phylogenetic analyses reveal HIV-1 infections between men misclassified as heterosexual transmissions. Aids. 2014; 28 (13):1967–1975. https://doi.org/10.1097/QAD.0000000000000383 PMID: 24991999

45. Bezemer D, Cori A, Ratmann O, van Sighem A, Hermanides HS, Dutilh BE, et al. Dispersion of the HIV- 1 Epidemic in Men Who Have Sex with Men in the Netherlands: A Combined Mathematical Model and Phylogenetic Analysis. PLoS Medicine. 2015; 12(11):e1001898. https://doi.org/10.1371/journal.pmed. 1001898 PMID: 26529093