The Kernel of Maximum Agreement Subtrees

Krister M. Swenson[1,3], Eric Chen[2],
Nicholas D. Pattengale[4], and David Sankoff[1]

[1]Department of Mathematics and Statistics, University of Ottawa, Ontario, K1N 6N5, Canada
[2]Department of Biology, University of Ottawa, Ontario, K1N 6N5, Canada
[3]LaCIM, UQAM, Montréal Québec, H3C 3P8, Canada
[4]Sandia National Laboratories, Albuquerque, New Mexico

Abstract. A Maximum Agreement SubTree (MAST) is a largest subtree common to a set of trees and serves as a summary of common substructure in the trees. A single MAST can be misleading, however, since there can be an exponential number of MASTs, and two MASTs for the same tree set do not even necessarily share any leaves. In this paper we introduce the notion of the Kernel Agreement SubTree (KAST), which is the summary of the common substructure in all MASTs, and show that it can be calculated in polynomial time (for trees with bounded degree). Suppose the input trees represent competing hypotheses for a particular phylogeny. We show the utility of the KAST as a method to discern the common structure of confidence, and as a measure of how confident we are in a given tree set.

1 Introduction

Phylogeny inference done on genetic data using maximum parsimony, maximum likelihood, and Bayesian analyses usually yields a set of most likely trees (phylogenies). A typical approach used by biologists to discern the commonality of the trees is to apply a consensus method which yields a single tree containing edges that are well represented in the set. For example, the majority-rules consensus tree contains only the edges (bipartitions of the leaf set) that exist in a majority of input trees. Consensus methods are also commonly used for their original purpose [1], to summarize the information provided from *different* data sets (there are other uses [32] but these are the two that we consider in this paper).

If one desires a more conservative summary, they may use the strict consensus tree, which has an edge if and only if the edge exists in all of the input trees. Yet even for this extremely conservative consensus method, there has been debate as to its validity and the conditions under which it should be used [3,23,4]. In particular, Barrett et al. [3] showed an example where a parsimony analysis of two data sets yields a consensus tree that is at odds with the tree obtained by combining the data. Nelson [23] replied with an argument that the error was not the act of taking the consensus, but the act of pooling the data.

The issue at the heart of this debate is, essentially, that of *wandering* or *rogue* leaves (taxa). Indeed, one or many leaves appearing in different locations of

J. Chen, J. Wang, and A. Zelikovsky (Eds.): ISBRA 2011, LNBI 6674, pp. 123–135, 2011.

otherwise identical trees have created the problems noticed by Barrett et al., and can also reduce the consensus tree to very few, if any, internal edges. On the other hand, Finden and Gordon [10] had already characterized Maximum Agreement SubTrees (MASTs): maximum cardinality subsets of the leaves for which all input trees agree. By calculating a MAST, one avoids Barrett's issue because all MASTs agree with the parsimonious tree they computed on the combined data. As we will see a single MAST can be misleading, however, as there can exist two MASTs (on a single set of trees) which share no leaves. Further, there are potentially an exponential (in the number of leaves) number of MASTs for a single set of trees [18]. For Barrett's example we will see that our new method appropriately excludes the contentious part of the tree, and so may be more fit than traditional consensus methods for comparing trees obtained from different analyses.

Wilkinson was the first to directly describe the issues surrounding rogue leaves and develop an approach to try to combat them [32]. Since then, a large body of work by Wilkinson and others has grown on the subjects of finding a single representative tree [32,33,34,31,11,24] or something other than a tree (forest, network, etc.) [2,17,9,15,26]. A full review of this work is out of the scope of this article so we refer the reader to the chapter of Bryant [8], the earlier work of Wilkinson [32,33], and Pattengale et al. [24]. Despite the myriad of options we notice a distinct lack of an efficiently computable base-line method for reporting subtrees of high confidence; a method analogous to the strict consensus, but less susceptible to rogue leaves. Thus, we introduce the Kernel Agreement SubTree (KAST) to summarize the information shared by all (potentially exponential) MASTs. Like the strict consensus, the KAST gives a summary of the common structure of high confidence, except that it excludes the rogue leaves that confound traditional consensus methods. The KAST has the benefits of having a simple definition, of summarizing the subtree of confidence by reporting a single tree, and unlike the other known subtree methods can be computed in polynomial time (when at least one input tree has bounded degree). Note that we do not use the term *kernel* in the machine learning sense (as in [28]).

When speaking of a reconstruction method that produces many most probable trees, Barrett et al.[3] called for "conservatism" and suggests the use of the strict consensus. In Section 5 we show the utility of the KAST as a means to get a conservative summary of many most probable trees. We then show the utility of the KAST in the original setting of consensus methods; on trees obtained through different analyses. In each setting we use the KAST not only to find subtrees of confidence, but as an indicator of randomness in the input trees.

The paper is organized as follows. We continue by formally defining the problem in Section 1.1 and showing properties of the MAST and KAST in Section 1.2. We then present Bryant's algorithm for computing the MAST in Section 2, on which our algorithm to compute the KAST (Section 3) is based on. Section 4 reports experimental values for the expected size of the KAST on various sets of trees generated at random while Section 5 shows how the KAST can be used to find subtrees of confidence, and report subsets of trees for which we are confident.

1.1 Definitions

Consider a set of trees $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ and a set of labels L such that each $x \in L$ labels exactly one leaf of each T_i. We will restrict a tree to a subset L' of its leaf set L; $T_i|_{L'}$ is the minimum homeomorphic subtree of T_i which has leaves L'. An *agreement subtree* for $\mathcal{T}$ is a subset $L' \subseteq L$ such that $T_1|_{L'} = T_2|_{L'} = \cdots = T_k|_{L'}$. A *maximum agreement subtree* (MAST) is an agreement subtree of maximum size. The set of all maximum agreement subtrees is $\mathcal{M}$.

Definition 1. *The* Kernel Agreement SubTree (KAST) *is the intersection of all MASTs (i.e.* $\cap_{T \in \mathcal{M}} T$*).*

See Figure 1 for an example.

As usual, node a is an *ancestor* of b if the path from b to the root passes through a. b is a *descendant* of a. For nodes a and b, the least common ancestor $lca(a, b)$ is the ancestor of a and b that is a descendant of all ancestors of a and b.

1.2 Properties of a MAST and the KAST

In Section 3 we show that the KAST can be computed in the same time as the fastest known algorithm to compute the MAST, by a convenient use of dynamic programming. The current fastest know algorithms for the MAST problem are due to Farach et al. [13] and Bryant [7]. Let d_i be the maximum degree (number of children) of tree $T_i \in \mathcal{T}$. These algorithms run in $O(kn^3 + n^d)$ time where $n = |L|$, k is the number of trees in the input, and d is the minimum over all d_i, $1 \leq i \leq k$.

We devote this section to showing desirable properties of the KAST by contrasting it with the MAST. First we look at the role KAST can play in Barrett's example [3]. The rooted trees obtained by his parsimony analyses are $T_1 = (A, (B, (C, D)))$ and $T_2 = (A, ((B, C), D)))$ (written in Newick format). The set of maximum agreement subtrees for T_1 and T_2 is

$$\{(A, (B, D)), (A, (D, C)), (A, (B, C))\}.$$

Thus, the KAST has only a single leaf A, which indicates that there is not enough information to imply a subtree of confidence. This is the result we would prefer to see, given the circumstances. We see more examples in Section 5 that show a KAST which finds substantial common substructure, yet does not falter by including subtrees that are at odds with biological observation.

Take a tree set $\mathcal{T}$ with a MAST of size m. Adding a tree T to $\mathcal{T}$ cannot result in a MAST larger than m. This is due to the fact that an agreement subtree of $\mathcal{T} \cup \{T\}$ must also be an agreement subtree of $\mathcal{T}$. On the other hand, the signal for a particular kernel can become apparent when more trees that agree are added to the set.

Property 1. The KAST on tree set $\mathcal{T}$ can be smaller than that of $\mathcal{T} \cup T$, for some tree T.

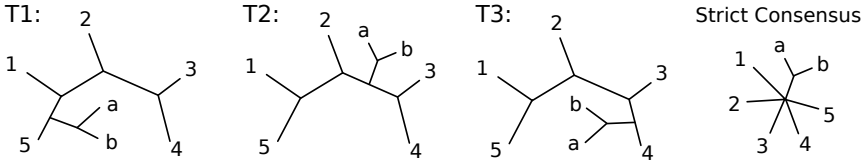


Fig. 1. The effect of adding a tree to the input set. The MASTs for $\{T_1, T_2\}$ are {1,2,3,4,5}, $\{a, b, 1, 3, 4\}$, $\{a, b, 2, 3, 4\}$, and $\{a, b, 3, 4, 5\}$, yielding the KAST $\{3, 4\}$. The MAST for $\{T_1, T_2, T_3\}$ is $\{1, 2, 3, 4, 5\}$, yielding the KAST $\{1, 2, 3, 4, 5\}$.

Figure 1 shows an example exhibiting this property. The KAST on input tree set $\{T_1, T_2\}$ has two leaves (is essentially empty) whereas the subtree on leaves $\{1, 2, 3, 4, 5\}$ is amplified by the addition of the tree T_3 to the set. We also see in Section 4 that the KAST size can often increase when adding somewhat similar trees to a set.

We finish with a few negative results about the MAST. The first shows that the MAST is not necessarily a good indicator of the common subtrees of confidence between two trees.

Property 2. There exists a family of tree sets that yields at least two MASTs, the intersection of which is size 2.

Take the caterpillar trees

$$(1, (2, (3, \ldots (n{-}1, n) \ldots))) \text{ and } (n/2, (n/2{+}1, \ldots, (n, (n/2{-}1, \ldots, (2, 1) \ldots)) \ldots))$$

for even n. Two of the MASTs for these trees are $\{1, 2, \ldots, n/2, n/2 + 1\}$ and $\{n/2, \ldots, n - 1, n\}$.

The second property shows that the number of MASTs and the size of them are not good indicators of their quality. We will see experimental evidence corroborating this fact in Section 4.

Property 3. There exists a family of tree sets that yields exactly two MASTs of size $\Omega(n)$, but the KAST is of size 4.

For this example we use trees that are nearly caterpillars. We write them as caterpillars, except S_1 denotes a subtree $(1a, 1b)$ while $S_{m/2+1}$ denotes a subtree $((m/2 + 1)a, (m/2 + 1)b)$. The first tree is then

$$(S_1, (2, (3, \ldots (S_{m/2+1}, \ldots (m - 1, m) \ldots)))$$

and the second is

$$(n, (n - 1, \ldots, (n/2 + 2, (S_1, (2, \ldots, (n/2, (S_{m/2+1})) \ldots))) \ldots))$$

where $m = n - 2$. The only two MASTs are now

$$\{1a, 1b, 2, \ldots, m/2, (m/2 + 1)a, (m/2 + 1)b\} \text{ and}$$
$$\{1a, 1b, (m/2 + 1)a, (m/2 + 1)b, m/2 + 2, \ldots, m - 1, m\}.$$

2 A Dynamic Programming Algorithm to Find the MAST

While either of the fastest known algorithms [7,13] for finding a MAST can be adapted to compute the KAST, we find it instructive to describe the algorithm of Bryant. We are comprehensive in our description. However, we refer the reader to Bryant's dissertation [7] for a more precise description of the algorithm.

Take $a, b \in L$ and call $\mathcal{T}(a, b)$ the set of all agreement subtrees where the $lca(a, b)$ is the root of the tree. Let $\mathcal{M}(a, b) \subseteq \mathcal{T}(a, b)$ be the set of maximum agreement subtrees where $lca(a, b)$ is the root, and $MAST(a, b)$ be the number of leaves in any member of $\mathcal{M}(a, b)$. We devote the rest of this section to computing $MAST(a, b)$ since the size of the MAST is simply the maximum $MAST(a, b)$ over all possible a and b.

Take three leaves $a, b, c \in L$. $ac|b$ denotes a *rooted triple* where $lca(a, c)$ is a descendant of $lca(a, b)$. In this case we say that c is *on a's side of the root* with respect to b (when $lca(a, b)$ is the root). Leaves a,b, and c form a *fan triple*, written (abc), if $lca(a, b) = lca(a, c) = lca(b, c)$. Define R to be the set of rooted triples common to all trees in $\mathcal{T}$ and F to be the set of fan triples common to all trees in $\mathcal{T}$. Bryant showed that an agreement subtree in $\mathcal{T}$ is equivalent to a subset of the set of rooted and fan triples.

The algorithm to compute $MAST(a, b)$ hinges upon the fact that the triples on a's side of the root, and the triples on b's side of the root can be addressed independently. Consider the set $X = \{x : xa|b \in R\} \cup \{a\}$ such that $lca(a, b)$ is the root. In this case, X corresponds to the leaves in a subtree on a's side of the root. Define $MAST_a = max\{MAST(a, x) : x \in X\}$ to be the MAST of the leaves in a subtree on a's side of the root. $MAST_b$ is defined similarly, where $X = \{x : a|bx \in R\} \cup \{b\}$.

If F is empty (i.e. the root of every tree in $\mathcal{T}$ is binary), then we have simply,

$$MAST(a, b) = MAST_a + MAST_b.$$

Otherwise, consider the maximum size subset $C \subseteq F$ such that $(abc) \in F$ for $c \in C$. Again, $MAST_c$ is the MAST that considers only the vertices x such that $xc|b$. The triples corresponding to some $MAST_c$ are not the same as those for $MAST_a$ and $MAST_b$. However, $MAST_c$ and $MAST_{c'}$ for $c, c' \in C$ could correspond to the same triples. To avoid conflict we construct a graph $G(C)$ as follows: for each $c \in C$ create a vertex with weight $MAST_c$. Make an edge between v and w if and only if $(bvw) \in F$ (i.e. v and w have the potential to appear in a subtree from the root that does not include a or b). A maximum weight clique S in this graph is the MAST of all potential subtrees that do not include a or b. So $MAST(a, b)$ can be written

$$MAST(a, b) = MAST_a + MAST_b + \sum_{s \in S} MAST_s$$

where $MAST_s$ is defined similarly to $MAST_a$ but with $X = \{x : a|sx \in R\} \cup \{s\}$.

3 Finding the KAST

$KAST(a, b)$ is the intersection of all MASTs in $\mathcal{M}(a, b)$ (the MASTs where $lca(a, b)$ is the root). In this section we show how to compute $KAST(a, b)$ through a modification of the algorithm of section 2.

Let $\mathcal{M}_a$ be the set of all MASTs on the leaf set $\{x : xa|b \in R\}$. In other words, $\mathcal{M}_a$ is the collection of sets of leaves that correspond to some $MAST_a$. Call $L(\mathcal{M}_a)$ the set of leaves in any MAST in $\mathcal{M}_a$ (i.e. $L(\mathcal{M}_a) = \{z \in M : M \in \mathcal{M}_a\}$). Symmetrically, $\mathcal{M}_b = \{x : a|bx \in R\}$ and $L(\mathcal{M}_b) = \{z \in M : M \in \mathcal{M}_b\}$. We begin by showing how to find $KAST(a, b)$ for binary trees.

Theorem 1. *If the trees $T_1, T_2, \ldots, T_k$ are binary, then*

$$KAST(a, b) = (\cap_{T \in \mathcal{M}_a} T) \cup (\cap_{T \in \mathcal{M}_b} T)$$

Proof. If $a = b$ then this is trivially true. Assume by induction that $KAST(c, d)$ can be calculated where $lca(a, b)$ is an ancestor of $lca(c, d)$.

Recall that $MAST(a, b) = MAST_a + MAST_b$ when the trees in $\mathcal{T}$ are binary and that $\mathcal{M}_a$ is the set of MASTs that include only the leaves a and x such that $lca(a, b)$ is an ancestor of $lca(a, x)$. It follows that $\mathcal{M}_a$ and $\mathcal{M}_b$ have the following property:

$$L(\mathcal{M}_a) \cap L(\mathcal{M}_b) = \emptyset$$

So $KAST(a, b)$ depends on $\mathcal{M}_a$ and $\mathcal{M}_b$ independently.

Bryant showed that any leaf included in $\mathcal{M}_a$ or $\mathcal{M}_b$ will necessarily exist in some MAST for $\mathcal{T}$ (a corollary of theorem 6.8 in [7]). Since the KAST contains only the leaves that exist in every MAST, then $KAST(a, b)$ must be equal to the intersection of all MASTs in $\mathcal{M}_a$. □

So the algorithm to compute $KAST(a, b)$ takes the intersection over all sets $KAST(c, d)$ such that $ac|b, ad|b \in R$ and $MAST(c, d)$ is maximum. It does the same for b's side of the root, and then takes the union of the result.

The following theorem hints that the independence of subsolutions that gives rise to MAST dynamic programming algorithms will similarly give rise to a KAST algorithm.

Theorem 2. *If any MAST is such that two leaves x, y are on the same side of the root, it follows that* every *MAST containing both x and y will also have them on the same side of the root.*

Proof. A simple proof by contradiction suffices. Assume that the theorem does not hold, namely that there is another MAST containing both x and y where x occurs on the other side of the root from y. Since the second MAST has root $lca(x, y)$ (because x and y are on either side of the root), this implies that the second MAST is a valid subtree in the first MAST, a contradiction. □

The implication here is that in building KAST subsolutions for one side of the root under consideration, we need not worry about leaves that we exclude being candidates for inclusion on the other side of the root.

We now present the main result of this section. Recall from Section 2 that $MAST(a,b) = MAST_a + MAST_b + \sum_{s \in S} MAST_s$ and the graph $G(C)$ where C is the set of triples satisfying $(abc) \in F$.

Theorem 3. *$KAST(a,b) = (\cap_{T \in \mathcal{M}_a} T) \cup (\cap_{T \in \mathcal{M}_b} T) \cup (\cap_{S \in \mathcal{K}} (\cup_{s \in S} (\cap_{T \in \mathcal{M}_s} T)))$ where $\mathcal{K}$ is the set of all maximum weight cliques on graph $G(C)$.*

Proof. If $a = b$ then this is trivially true. Assume by induction that $KAST(c,d)$ can be calculated where $lca(a,b)$ is an ancestor of $lca(c,d)$.

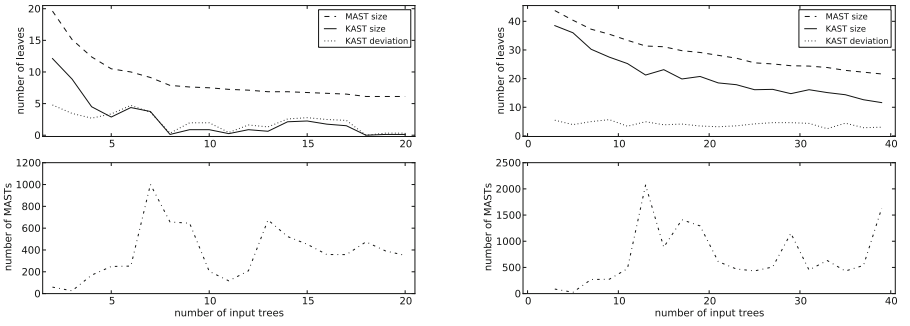
Take any maximum weight clique $S \in \mathcal{K}$. Bryant showed that for $S = \{s_1, \ldots, s_m\}$, $\cup_{i=1}^{m} T_i$ where $T_i \in \mathcal{M}_{s_i}$, is a MAST on the set of leaves $\{c : (abc) \in C\}$. By the definition of $G(C)$ we know that $L(\mathcal{M}_{s_1})$, $L(\mathcal{M}_{s_2})$,...,$L(\mathcal{M}_{s_m})$,$L(\mathcal{M}_a)$, and $L(\mathcal{M}_b)$ are pairwise disjoint. Further, any leaf in the sets $L(\mathcal{M}_{s_i})$, $L(\mathcal{M}_a)$, or $L(\mathcal{M}_b)$ are necessarily included in some MAST for $\mathcal{T}$ (a corollary of theorem 6.8 in [7]). So the leaves in a $KAST(a,b)$ could have only the leaves that are in every MAST in $\mathcal{M}_{s_i}$ (i.e. $(\cup_{s \in S} (\cap_{T \in \mathcal{M}_s} T))$), for all $1 \leq i \leq m$. But each clique in $\mathcal{K}$ represents a different MAST, so only the leaves that are in every clique will be in the KAST. Finally, this set is disjoint from $L(\mathcal{M}_a)$ and $L(\mathcal{M}_b)$ for the same reason that $L(\mathcal{M}_a)$ and $L(\mathcal{M}_b)$ are disjoint from each other. □

4 Experiments

We implemented the KAST from code that computes the MAST in the phylogenetic package RAxML [29]. In this section we report empirical evidence about the expected size of the KAST and MAST under two different models. The first model builds a tree set $\mathcal{T}$ of random trees constructed through a birth/death process, while the second starts with a random birth/death tree and then produces new trees by doing Nearest Neighbor Interchange (NNI) moves [27,22]. This way we see how the expected sizes react to adding drastically dissimilar, or fairly similar trees to the set $\mathcal{T}$. In Figure 2, we show that as the size of $\mathcal{T}$ (the tree set) increases, the size of the KAST decreases precipitously in the case of the birth/death model, whereas it decreases more gracefully in the case of the NNI model. Each plot is generated from an initial birth/death tree on 50 leaves, where new trees are added to the tree set according to the prescribed model. This process was repeated 10 times and the average is reported. Plots with various numbers of leaves are similar except that the curve is scaled on the "leaves" axis proportionately. In regards to the number of MASTs, the plots show an erratic curve, confirming that the phenomenon described in Property 3 is not a rarity.

5 Applications

We now demonstrate the application of the KAST in finding subtrees of confidence, as well as finding subsets of the input tree set of confidence. To do this we gleaned phylogenies from the literature that are known to have an agreed upon



(a) random birth/death trees *(b) random NNI moves*

Fig. 2. Expected values of the MAST and KAST sizes

structure, except for a few contentious leaves. Our intention is not to provide biological insight, but to confirm the utility of the KAST by comparing our results to familiar phylogenies. The real utility of the KAST will be on phylogenies that are much larger, so large as to make it difficult for a humans to process.

5.1 Analyses on Flatworm Phylogenies

In a recent publication by Philippe et al.[25], the proposed phylogeny describes the Acoel and the Nemertodermatids and Xenoturbellid as a sister-clade to Ambulacraria, which is vastly different from the previous publications. The competing hypotheses are depicted in Figure 3. In earlier publications both Nemertodermatids and Acoels are the outgroups with Xenoturbellid leaf grouping either with the Ambulacraria or with the Nemertodermatids and Acoels. Setting aside the interpretation and biological ramifications of the new proposed tree topology, it is a good real-world example for observing the effects of KAST on contentious trees.

There are two main objectives that we wish to explore through the use of this example. The first objective is to determine if the kernel of a set of phylogenetic trees can identify a subset that we are confident in. The second objective is to show the KAST as a measurement of how confident we are in the hypothesis of these trees.

Confidence in the phylogenetic tree. From their publication, Philippe et al.[25] presents three trees, two from prior publications and one from their own experimental result. With only 10 common leaves among the three trees it is very easy to identify the similarity between them by eye (Figure 3).

A quick observation can find that Ecdysozoans and Lophotrochozoans of Protostomia forms a clade, Vetebrates and Urochordates and Cephalochordates of Chordata forms a clade, and Hemichordates and Echinoderms of Ambulacraia forms a sister clade to Chordata; and that Xenoturbellid, Nemertodermatids, and Acoels are the rogue leaves. The KAST of these three trees agrees with this

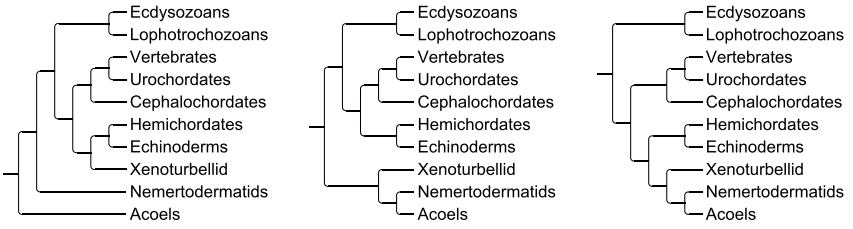


Fig. 3. Phylogenies from Figure 1 of Philippe et al.[25] Xenoturbellid, Nemertodermatids, and Acoels wander

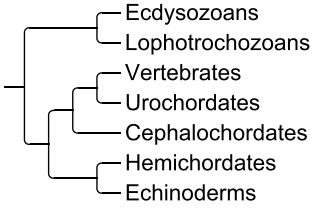


Fig. 4. The KAST of phylogenies of Figure 3

observation (Figure 4). This suggests that the KAST is able to find a subtree that is not only biologically obvious but also likely to have significant support.

With bigger trees it will be harder to identify the similarity. We would argue that the KAST can be an important tool in identifying or verifying these similarities.

Phylogeny reconstruction. To find if the kernel in a set of phylogenies could identify a subset of trees that we are confident of in the context of phylogeny reconstruction, we tried to replicate the analysis of Philippe et al.[25] The aligned mitochondrial gene set was taken from the supplementary material section and used as input for the Bayesian analysis that they used: PhyloBayes 3.2[20,19], with the CAT model[25,20] as the amino acid replacement model and default settings for everything else. We ran 10000 cycles and discarded the 1000 burn-ins, as they did. The consensus tree by majority rule was then obtained by CONSENSE [14] using all remaining 9000 trees. The consensus tree (Figure 5) we found is in agreement with the CAT + Γ model tree from their supplementary material Figure 1.

To test the validity of the conservative tree produced by the KAST, the kernel of the 9000 trees set is calculated. Of the 10 species in the KAST, three sponge species and two jellyfish species group together as predicted, the two annelida species group together as predicted, the three echinoderms also group together as predicted, and the topology of these phyla are also organized in the biologically obvious fashion (Figure 6). This corroborates the notion that the topology of KAST is the base-line topology.

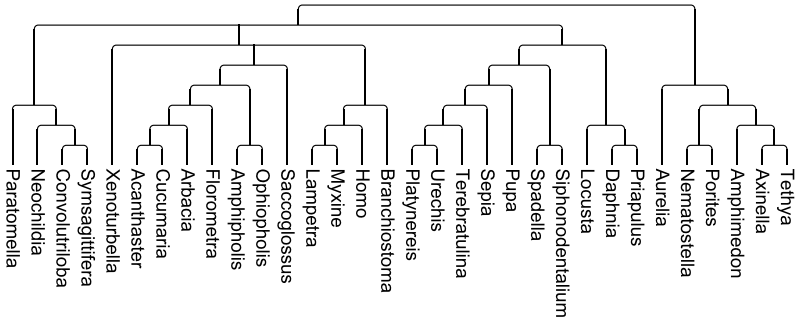


Fig. 5. The majority rule consensus tree using 9000 trees. The topology is essentially the same as Supplementary Figure 1 from Philippe et al.[25]

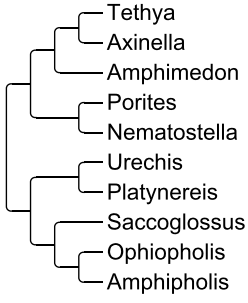


Fig. 6. The KAST of the 9000 trees from the Baysian analysis program PhyloBayes

Next we test the variability of the KAST within the tree set. Philippe et al.[25] sampled once every 10 cycles, to simulate this we sample 900 random trees in the 9000 tree set and calculate the KAST. We replicate this 1000 times and calculate the symmetric Robinson-Foulds distances (because the KAST is binary, we divide it by two) between every pair of KASTs generated. The average distance between these KASTs is 0.73 with an average KAST size of 10.73.

We also calculate the size of KAST with varying numbers of tree sets to test how sample size effects the KAST size. Samples of 5000, 2500, 1200, and 500 trees all have KAST size of 11. Starting with the samples of 250 trees the size of KAST start to increase, with samples of 30 trees having a KAST size of 16. While the KAST from the whole 9000 trees set is obviously more conservative, the KAST from the smaller samples agree with all known competing hypotheses while including up to half the leaves.

5.2 Analyses on γ-Proteobacteria Phylogenies

Finally, we test the KAST on the phylogeny of γ-proteobacteria that has been the subject of pains-taking study. We refer the reader to Herbeck et al. [16]

for a discussion of previous work. For our purposes, we concentrate on the studies related to 12 particular species used in Lerat et al. [21], who reconstructed a phylogeny based on hundreds of genes. Since then there have been other attempts to reconstruct the phylogeny based on the syntenic data of the whole genome[12,6,5,30].

We turn our attention to two studies that produced trees in discordance with that of Lerat. Belda et al. [5] produced two trees, one using Maximum Likelihood on amino acid sequences and the other using reversal distance on the syntenic information (they used the breakpoint distance as well, which produced the same tree as the inversion distance). The likelihood analysis gave a tree that agreed with Lerat's. The inversion distance gave a tree that has significant differences to that of Lerat; the KAST between the two has 9 of 12 leaves. However, we will see that when we add certain trees from the study of Blin et al., the KAST size is 10. Further, the leaves excluded are Wigglesworthia brevipalpis and Pseudomonas aeruginosa; the former identified by Herbeck et al. [16] as troublesome to place, and the latter being the outgroup that they used to root their trees.

Blin et. al [6] used model free distances (breakpoints, conserved intervals, and common intervals) on the syntenic data to reconstruct their phylogenies. They produced many trees with the various methods on two different data sets. The syntenic data that yielded the interesting phylogeny for our purposes was produced from coding genes along with ribosomal and transfer RNAs. Blin et al. noticed that their trees computed on this data, using conserved and common intervals, were more different from the Lerat tree than the others. The KAST confirms this: the KAST on the set of all published trees other than these two is 10 while the inclusion of either one (they are the same) yields a KAST of size 5. Our experimental data tells us that a sequence of six trees, each produced by a random NNI operation from the last, will yield a KAST of size 5 while six unrelated trees would produce a KAST of size 2. We conclude that we have higher confidence in the set of trees that don't include those two trees.

6 Conclusion

We claim that the utility of the KAST is two-fold. The first is that the KAST is a safe summary of the subtree of confidence for a set of trees. The second is that the size of the KAST is correlated with how related the set of trees is. The KAST is not as susceptible to rogue taxa as the very conservative strict consensus, and is not as misleading as the MAST can be. Furthermore, unlike the other methods that attempt to characterize structure in the presence of rogue taxa, our measure is computable in polynomial time.

Acknowledgments

The first author would like to thank Andre Aberer for use and discussions about his code to compute the MAST, and Bernard Moret for discussions about the kernel agreement subtree.

References

1. Adams, E.N.: Consensus techniques and the comparison of taxonomic trees. Syst. Zool. 21, 390–397 (1972)
2. Bandelt, H., Dress, A.: Split decomposition: A new and useful approach to phylogenetic analysis of distance data. Mol. Phyl. Evol. 1(3), 242–252 (1992)
3. Barrett, M., Donoghue, M.J., Sober, E.: Against consensus. Syst. Zool. 40(4), 486–493 (1991)
4. Barrett, M., Donoghue, M.J., Sober, E.: Crusade? a reply to Nelson. Syst. Biol. 42(2), 216–217 (1993)
5. Belda, E., Moya, A., Silva, F.J.: Genome rearrangement distances and gene order phylogeny in γ-proteobacteria. Mol. Biol. Evol. 22(6), 1456–1467 (2005)
6. Blin, G., Chauve, C., Fertin, G.: Genes order and phylogenetic reconstruction: Application to γ-proteobacteria. In: Lagergren, J. (ed.) RECOMB-WS 2004. LNCS (LNBI), vol. 3388, pp. 11–20. Springer, Heidelberg (2005)
7. Bryant, D.: Building trees, hunting for trees, and comparing trees. PhD dissertation, Department of Mathematics, University of Canterbury (1997)
8. Bryant, D.: A classification of consensus methods for phylogenetics. In: Bioconsensus. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 61, pp. 163–184. AMS Press, New York (2002)
9. Bryant, D., Moulton, V.: Neighbor-net: an agglomerative method for the construction of phylogenetic networks. Mol. Biol. Evol. 21(2), 255–265 (2004)
10. Gordon, A.D., Finden, C.R.: Obtaining common pruned trees. J. Classification 2(1), 255–267 (1985)
11. Cranston, K.A., Rannala, B.: Summarizing a posterior distribution of trees using agreement subtrees. Syst. Biol. 56(4), 578–590 (2007)
12. Earnest-DeYoung, J.V., Lerat, E., Moret, B.M.E.: Reversing gene erosion – reconstructing ancestral bacterial genomes from gene-content and order data. In: Jonassen, I., Kim, J. (eds.) WABI 2004. LNCS (LNBI), vol. 3240, pp. 1–13. Springer, Heidelberg (2004)
13. Farach, M., Przytycka, T., Thorup, M.: On the agreement of many trees. Information Processing Letters, 297–301 (1995)
14. Felsenstein, J.: Phylogenetic Inference Package (PHYLIP), Version 3.5. University of Washington, Seattle (1993)
15. Gauthier, O., Lapointe, F.-J.: Seeing the trees for the network: consensus, information content, and superphylogenies. Syst. Biol. 56(2), 345–355 (2007)
16. Herbeck, J.T., Degnan, P.H., Wernegreen, J.J.: Nonhomogeneous model of sequence evolution indicates independent origins of primary endosymbionts within the enterobacteriales (*gamma*-proteobacteria). Mol. Biol. Evol. 22(3), 520–532 (2005)
17. Huson, D.H.: SplitsTree: analyzing and visualizing evolutionary data. Bioinformatics 14(1), 68–73 (1998)
18. Kubicka, E., Kubicki, G., McMorris, F.R.: On agreement subtrees of two binary trees. Congressus Numeratium 88, 217–224 (1992)
19. Lartillot, N., Brinkmann, H., Philippe, H.: Suppression of long-branch attraction artefacts in the animal phylogeny using a site-heterogeneous model. BMC Evol. Biol. 7(Suppl. 1) (2007); 1st International Conference on Phylogenomics, St Adele, CANADA, March 15-19 (2006)
20. Lartillot, N., Philippe, H.: A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. Mol. Biol. Evol. 21(6), 1095–1109 (2004)

21. Lerat, E., Daubin, V., Moran, N.A.: From gene trees to organismal phylogeny in prokaryotes:the case of the γ-proteobacteria. PLoS Biol. 1(1), e19 (2003)
22. Moore, G.W., Goodman, M., Barnabas, J.: An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets. Journal of Theoretical Biology 38(3), 423–457 (1973)
23. Nelson, G.: Why crusade against consensus? a reply to Barret, Donoghue, and Sober. Syst. Biol. 42(2), 215–216 (1993)
24. Pattengale, N.D., Aberer, A.J., Swenson, K.M., Stamatakis, A., Moret, B.M.E.: Uncovering hidden phylogenetic consensus in large datasets. IEEE/ACM Transactions on Computational Biology and Bioinformatics 99(PrePrints) (2011)
25. Philippe, H., Brinkmann, H., Copley, R.R., Moroz, L.L., Nakano, H., Poustka, A.J., Wallberg, A., Peterson, K.J., Telford, M.J.: Acoelomorph flatworms are deuterostomes related to Xenoturbella. Nature 470(7333), 255–258 (2011)
26. Redelings, B.: Bayesian phylogenies unplugged: Majority consensus trees with wandering taxa, `http://www.duke.edu/~br51/wandering.pdf`
27. Robinson, D.F.: Comparison of labeled trees with valency three. Journal of Combinatorial Theory, Series B 11(2), 105–119 (1971)
28. Shin, K., Kuboyama, T.: Kernels based on distributions of agreement subtrees. In: Wobcke, W., Zhang, M. (eds.) AI 2008. LNCS (LNAI), vol. 5360, pp. 236–246. Springer, Heidelberg (2008)
29. Stamatakis, A.: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22(21), 2688–2690 (2006)
30. Swenson, K.M., Arndt, W., Tang, J., Moret, B.M.E.: Phylogenetic reconstruction from complete gene orders of whole genomes. In: Proc. 6rd Asia Pacific Bioinformatics Conf. (APBC 2008), pp. 241–250 (2008)
31. Thorley, J.L., Wilkinson, M., Charleston, M.: The information content of consensus trees. In: Rizzi, A., Vichi, M., Bock, H. (eds.) Studies in Classification, Data Analysis, and Knowledge Organization, Advances in Data Science and Classification, pp. 91–98. Springer, Heidelberg (1998)
32. Wilkinson, M.: Common cladistic information and its consensus representation: reduced adams and reduced cladistic consensus trees and profiles. Syst. Biol. 43(3), 343–368 (1994)
33. Wilkinson, M.: More on reduced consensus methods. Syst. Biol. 44, 435–439 (1995)
34. Wilkinson, M.: Majority-rule reduced consensus trees and their use in bootstrapping. Mol. Biol. Evol. 13(3), 437–444 (1996)