A Quartet-Based Approach to Refining Polytomies

Sean Liu

Apr 2025

1 Introduction

Quartet-based method to infer phylogenetic trees have been shown to have desirable theoretical guarantees under various models of evolution, and have shown great promise in empirical performance. However, finding the tree that satisfies the most quartets in some multiset of quartets is NP-Hard, and while algorithms exist to approximately solve this problem, they still exhibit superlinear scaling, prohibiting quartet-based analyses of large (in the number of taxa) datasets.

For example, in the case of phylogenetic analyses of linguistic data, IE-CoR has 161 taxa, and when quartets are generated under the EVANS-ONE scheme, ASTRAL-III [2] runs into both memory (requiring more than 512 GB of RAM) and compute constraints. However, we may utilise uncontroversial linguistic information to speed up our search. In particular, known language groups (for example, the existence of the Germanic clade) and subgroups (East, North, and West Germanic) provide information that may be cast in terms of an unrefined "backbone" tree. Hence we may restrict our search space to the space of all refinements of the backbone tree. Quartets that conflict with the backbone tree or are compatible with every refinement of that backbone tree may be ignored, and of the rest of the quartets, they can be informative to at most one polytomy in the guide tree. Using this observation a divide-and-conquer algorithm can be devised that resolves each polytomy independently (lending itself to parallelisation) and then stitches them back together.

Note: Rabiee & Mirarab have already tackled this problem in [1], though their approach was to modify the constraint bipartition set of ASTRAL in a sophisticated way; this method is more naïve but lends itself well to parallisation and extension as it does not rely on any specific quartet-based inference software.

2 Quartet Perspective

Definition 1 *Here we define how ASTRAL computes* ***quartet edge support****. Suppose $q = ab|cd$ is a quartet and e is some edge on a tree T. Then q* ***supports*** *e only if e separates T into four quadpartitions, A, B, C, D such that $a \in A$,*

*$b \in B$, $c \in C$, $d \in D$, and A, B are separated by e from C, D. Whenever a quartet supports an edge on T, it is **compatible** with T.*

I will now define the Maximum Quartet Support Species Tree with Clade Constraints (MQSST+CC) problem: given a leafset X, a set of clades Y, a clade-mapping function $f : X \rightarrow Y$, and a set of quartets Q, one would like to find an unrooted binary tree T such that the maximum number of quartets in Q are compatible with T *and* the leaves are sorted according to the clade constraints. More precisely, for each $x \in X$, there exists an edge e in T such that all the nodes on the same side as x after removing e has the same clade; that is, if $L|R$ is the bipartition on the leafset obtained by removing e from T and $x \in L$, then for all $x' \in L$ we have $f(x) = f(x')$.

2.1 Divide-and-conquer tree search strategy

I will now show that this problem can be solved in three steps using divide-and-conquer in three steps: (1) finding the unrooted "backbone tree" of clades, for every clade, finding (2) the rooted tree on all taxa in that clade, and then finally (3) attaching the clades back to the rooted tree.

Suppose that $ab|cd \in Q$, and that $f(a) = A, f(b) = B, f(c) = C$, and $f(d) = D$. Now, consider the "family quartet" $AB|CD$ induced by $ab|cd$. There are only 7 different possible patterns for this family quartet depending on which taxa belong to the same families, up to quartet equivalence. That is, $ab|cd$ is equivalent to $ba|cd$, $ab|dc$, and $cd|ab$. In particular, some quartets are **incompatible** (which means that the quartet is incompatible with the given clade constraints; no tree satisfying the clade constraints will be compatible with the quartet) and some are **uninformative** (which means that the quartet is compatible with *every* tree that satisfies the clade constraints). As it turns out, only 1 pattern is useful for step (1) and only two patterns are useful for step (2); the rest can be thrown out. All possible patterns are enumerated in Table 1.

What happens in step 1 is thus very easy: take only those quartets whose four taxa all belong to different families and find the tree that agrees with the most quartets - for example, using ASTRAL.

But step 2 requires a little more care: as a start, it estimates *rooted* phylogenies which require an outgroup. Thus we have to add an outgroup taxon to each family to perform inference - we will call that taxon x_A for family A. Obviously, there is no information about the outgroup to be inferred from quartets whose taxa all belong to the same family. Thus the only information can come from those of the form $AA|AB$, where A is the current family and B is some other family. Suppose that originally this quartet was $ab|cd$, where $f(a) = f(b) = f(c) = A$ and $f(d) = B$. Then replace d by the outgroup node of the family x_A to obtain the taxon $ab|cx_A$. Thus for a family A, one can take all quartets with at least three taxa in the family, replace all taxa in quartets that are not in A with the outgroup taxon x_A, and again use ASTRAL to infer a tree on this group.

Table 1: **All possible patterns of family quartets.** For some quartet $ab|cd$, we consider what happens if we replace each taxa with the families that they belong to. The difference between such resulting "family quartets" and normal quartets is that families may repeat in each quartet. Under given family constraints this would make some quartets still informative while rendering others uninformative or incompatible. This table enumerates all possible family quartet patterns up to quartet equivalence. For example, the pattern $AA|BB$ is the same as $BB|AA$ and is not repeated.

No. distinct families	Family Quartet Pattern	Comment
4	$AB\|CD$	Used in step 1
3	$AB\|AC$	Uninformative
	$AA\|BC$	Incompatible
2	$AA\|BB$	Uninformative
	$AB\|AB$	Incompatible
	$AA\|AB$	Used in step 2
1	$AA\|AA$	Used in step 2

Hence this leads to a simple divide-and-conquer algorithm for the maximum quartet compatibility with clade constraints tree problem: the input is a set of quartets Q on some leafset X, a set of families Y, and a mapping $f : X \to Y$ that maps taxa to the family. The output is a tree T such that it satisfies the clade constraints and satisfies the maximum number of quartets in Q. We assume that there is an oracle for the maximum quartet compatibility tree problem $\textsc{MaxQuartetComp}(Q)$.

1. Backbone tree: let Q_{bb} be the set of quartets in Q whose four taxa all belong to different families. Let $T_{\text{bb}} = \textsc{MaxQuartetComp}(Q_{\text{bb}})$; its leafset is Y.

2. For each family A: let Q_A be the set of quartets in Q which are informative for A. That is, if a quartet has all four taxa in A, it goes into Q_A unmodified; otherwise if it has exactly one taxa not in A, replace that taxa with the outgroup taxon x_A and insert it into Q_A. Then let $T_A = \textsc{MaxQuartetComp}(Q_A)$ be the rooted tree on the family of A.

3. Conquer step: For each leaf $F \in Y$ in T_{bb}, attach T_F to T_{bb} by joining F in T_{bb} and x_F in T_F. Delete both x_F and F. Call this tree T_{res}.

4. Return T_{res}.

Reduction to ASTRAL when solving in exact mode There is in fact a way of doing this without divide-and-conquer using ASTRAL which has to do with the input bipartitions. This is equivalent to (1) removing all quartets that are either incompatible or uninformative in Table 1, and then (2) feeding ASTRAL the correct set of bipartitions $\mathcal{X}$, constructed as follows: for each family $f \in Y$, let $\text{tax}(f)$ denote the set of taxa that are in that family. For each $S \subseteq \text{tax}(f)$ insert the bipartition $S|X - S$ into $\mathcal{X}$. This corresponds to step 2. For each subset of families $U \subseteq Y$, let $\text{tax}(U) = \cup_{f \in U} \text{tax}(f)$ denote the set of taxa whose families are in U. Then insert $\text{tax}(U)|X - \text{tax}(U)$ into $\mathcal{X}$. Finally, run ASTRAL on the set of quartets obtained in step (1) with the constraint bipartitions as defined in step (2). This will solve the problem exactly, but may be computationally heavy - if there are m families each of size n, then $|\mathcal{X}| = O(m2^n + 2^m)$. On the other hand, this is much faster than running ASTRAL in exact mode directly on the set of $O(mn)$ taxa, as the number of bipartitions then would be $O(2^{mn})$. Note that step (1), the quartet filtering step, isn't even strictly necessary, so that all you *really* need to do is to compute the new bipartitions.

2.2 The Maximum Quartet Support Species Refinement Tree Problem

Now, I will state a natural generalisation of the MQSST+CC problem: the Maximum Quartet Support Species Refinement Tree (MQSSRT) problem. In short, the difference is that now the constraint is a tree T such that the output tree T^* has to be a refinement of T. It is not hard to see that having clade constraints is just a special case of this problem. Equivalently, one needs to resolve all the polytomies in T so that the resulting tree has maximum total quartet support.

The formal description is thus: Given a tree T and a set of quartets Q, find a refinement of T such that the tree is compatible with the most number of quartets in Q. This problem can be solved by refining every polytomy, as follows: suppose that there is a polytomy at u. We will build a set of quartets $Q_u \subseteq Q$ such that running a quartet method on Q_u resolved the polytomy at u. Gather up all quartets $ab|cd \in Q$ such that u splits a, b, c, d into four distinct connected components when removed from T. Then suppose A is on the path from a to u, and that u is linked by an edge to A, and let Nodes B, C, D be found similarly (that is, B is the node such that u is connected to B and it lies on the path from b to u). Then add the quartet $AB|CD$ to Q_u. Finally, run ASTRAL on all the sets in Q_u to obtain a resolution of the polytomy at u. Then just stitch all polytomies and non-polytomies together for the final tree. Polytomies can be resolved in parallel because they are independent problems, so the runtime (if assuming max. parallelism), is $O(2^{\Delta})$, where Δ is the maximum degree of T.

References

[1] Maryam Rabiee and Siavash Mirarab. Forcing external constraints on tree inference using ASTRAL. *BMC genomics*, 21:1–13, 2020.

[2] Chao Zhang, Maryam Rabiee, Erfan Sayyari, and Siavash Mirarab. ASTRAL-III: polynomial time species tree reconstruction from partially resolved gene trees. *BMC bioinformatics*, 19:15–30, 2018.