# Maximum Compatible Tree

Vincent Berry

# Maximum Compatible Tree (2001; Ganapathy,Warnow)

Vincent Berry, LIRMM, UNIV. MONTPELLIER II – CNRS, www.lirmm.fr/~vberry
entry editor: Vincent Berry

**INDEX TERMS:** Trees, Phylogenetics, Maximum Compatible Tree, Pattern Matching on Trees, Consensus of Trees.
**SYNONYMS:** maximum refinement subtree (MRST).

## 1 PROBLEM DEFINITION

This problem is a pattern matching problem on leaf-labeled trees. Each input tree is considered as a branching pattern inducing specific groups of leaves. Given a set of input trees with identical leaf sets, the goal is to find a largest subset of leaves on the branching pattern of which the input trees do not disagree. A *maximum compatible tree* is a tree with such a leaf-set and with the branching patterns associated to these leaves by the input trees. The Maximum Compatible Tree problem (MCT) is to find such a tree or, equivalently, its leaf set. The main motivation for this problem is in phylogenetics, to measure the similarity between evolutionary trees, or to represent a consensus of a set of trees. The problem was introduced in [9] and [10, under the MRST acronym]. Previous related works concern the well-known Maximum Agreement Subtree problem (MAST). Solving MAST is finding a largest subset of leaves on which all input trees *exactly* agree. The difference between MAST and MCT, is that MAST seeks a tree whose branching information is isomorphic to that of a subtree in each of the input trees, while MCT seeks a tree that contains the branching information (*i.e.* groups) of a subtree of each input tree. This difference allows the tree obtained for MCT to be more informative, as it can include branching information present in one input tree but not in the others, as long as this information is compatible with them. Both problems are equivalent when all input trees are binary. Ganapathy and Warnow [5] were the first to give an algorithm to solve MCT in its general form. Their algorithm relies on a simple dynamic programming approach similar to a work on MAST [12] and has a running time exponential in the number of input trees and in the maximum degree of a node in the input trees. Later, [2] proposed a fixed-parameter algorithm using one parameter only. Approximation results have also been obtained [1, 6], the result being low-cost polynomial-time algorithms that approximate the complement of MCT within a constant threshold.

**Notations** Trees considered here are evolutionary trees (*phylogenies*). Such a tree $T$ has its leaf set $L(T)$ in bijection with a label set and is either rooted, in which case all internal nodes have at least two children each, or unrooted, in which case internal nodes have a degree of at least three. Given a set $L$ of labels and a tree $T$, the *restriction* of $T$ to $L$, denoted $T|L$, is the tree obtained in the following way: take the smallest induced subgraph of $T$ connecting leaves with labels in $L \cap L(T)$, then remove any degree two (non-root) node to make the tree homeomorphically irreducible. Two trees $T$, $T'$ are *isomorphic*, denoted $T = T'$, if and only if there is a graph isomorphism $T \mapsto T'$ preserving leaf labels (and the root if both trees are rooted). A tree $T$ *refines* a tree $T'$, denoted $T \trianglerighteq T'$, whenever $T$ can be transformed into $T'$ by collapsing some of its internal edges (*collapsing* an edge means removing it and merging its extremities). See Figure 1 for examples of these relations between trees. Note that a tree $T$ properly refining another tree $T'$, agrees with the entire

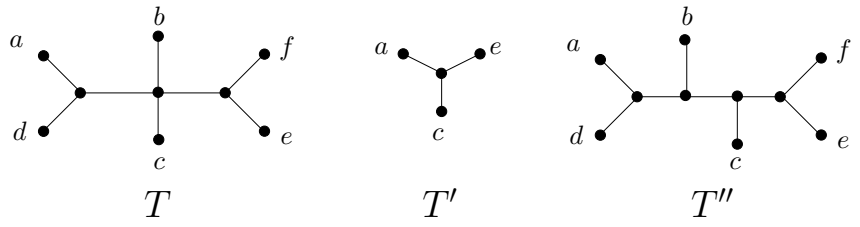


Figure 1: Three unrooted trees. A tree $T$, a tree $T'$ such that $T' = T|\{a, c, e\}$ and a tree $T''$ such that $T'' \trianglerighteq T$.

evolutionary history of $T'$, while containing additional information absent from $T'$: at least one high degree node of $T'$ is replaced in $T$ by several nodes of lesser degree, hence $T$ contains more speciation events than $T'$. Given a collection $\mathcal{T} = \{T_1, T_2, \ldots, T_k\}$ of input trees with identical leaf sets $L$, a tree $T$ with leaves in $L$ is said to be *compatible with* $\mathcal{T}$ if and only if $\forall T_i \in \mathcal{T}$, $T \trianglerighteq T_i|L(T)$. If there is a tree $T$ compatible with $\mathcal{T}$ such that $L(T) = L$, then the collection $\mathcal{T}$ is said to be *compatible*. Knowing whether a collection is compatible is a problem for which linear-time algorithms have been known for a long time (*e.g.* [8]). The Maximum Compatible Tree problem is a natural optimization version of this problem to deal with incompatible collections of trees.

**Problem 1** (Maximum Compatible Tree – mct)**.**
Input*: A collection $\mathcal{T}$ of trees with the same leaf sets.*
Output*: A tree compatible with $\mathcal{T}$ having the largest number of leaves. Such a tree is denoted $MCT(\mathcal{T})$.*

See Figure 2 for an example. Note that $\forall \mathcal{T}$, $|MCT(\mathcal{T})| \geq |MAST(\mathcal{T})|$ and that mct is equivalent to mast when input trees are binary. Note also that instances of mct and mast can have several optimum solutions.

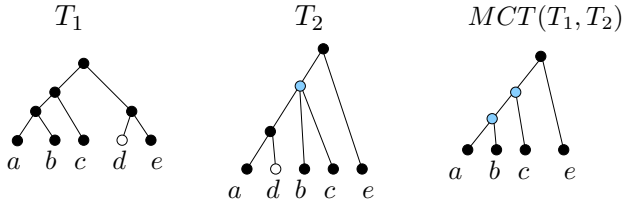


Figure 2: An incompatible collection of two input trees $\{T_1, T_2\}$ and their maximum compatible tree, $T = MCT(T_1, T_2)$. Removing the leaf $d$ renders the input trees compatible, hence $L(T) = \{a, b, c, e\}$. Here, $T$ strictly refines $T_2$ restricted to $L(T)$, which is expressed by the fact that a node in $T_2$ (the blue one) has its child subtrees distributed between several connected nodes of $T$ (blue nodes). Note also that here $|MCT(T_1, T_2)| > |MAST(T_1, T_2)|$.

# 2 KEY RESULTS

## Exact algorithms

The mct problem was shown to be NP-hard on 6 trees in [9], then on 2 trees in [10]. The NP-hardness holds as long as one of the input trees is not of bounded degree. For two bounded-degree trees, Hein et al. mention a polynomial-time algorithm based on *aligning* trees. The work of

Ganapathy and Warnow [5] proposes an exponential algorithm for solving MCT in the general case. Given two trees $T_1, T_2$, they show how to compute a binary MCT of any pair of subtrees ($S_1 \in T_1, S_2 \in T_2$) by dynamic programming. Subtrees whose root is of high degree are handled by considering every possible partition of the roots's children in two sets. This leads the complexity bound to have a term exponential in $d$, the maximum degree of a node in the input trees. When dealing with $k$ input trees, $k$-tuples of subtrees are considered, and the simultaneous bipartitions of the roots's children for $k$ subtrees are considered. Hence, the complexity bound is also exponential in $k$.

**Theorem 1.** *[5] Let $L$ be a set of $n$ leaves. The* MCT *problem for a collection of $k$ rooted trees on $L$ in which each tree has degree at most $d+1$, can be solved in $O(2^{2kd}n^k)$ time.*

The result easily extends to unrooted trees by considering each of the $n$ leaves in turn as a possible root for all trees of the collection.

**Theorem 2.** *[5] Given a collection of $k$ unrooted trees with degree at most $d+1$ on an $n$-leaf set, the* MCT *problem can be solved in $O(2^{2kd}n^{k+1})$.*

Let $\mathcal{T}$ be a collection on a leaf-set $L$, [2] considered the following decision problem, denoted MCT$_p$: given $\mathcal{T}$ and $p \in [0, n]$, does $|MCT(\mathcal{T})| \geq n - p$?

**Theorem 3.** *[2]*

1. MCT$_p$ *on rooted trees can be solved in $O(\min\{3^p kn, 2.27^p + kn^3\})$ time.*
2. MCT$_p$ *on unrooted trees can be solved in $O\big((p+1) \times \min\{3^p kn, 2.27^p + kn^3\}\big)$ time.*

The $3^p kn$ term comes from an algorithm that first locates in $O(kn)$ time a 3-leaf set $S$ on which the input trees conflict, then recursively obtains a maximum compatible tree $T_1$, resp. $T_2$, $T_3$ for each of the three collecions $\mathcal{T}_1$, resp. $\mathcal{T}_2, \mathcal{T}_3$ obtained by removing from the input trees a leaf in $S$, and last returning the $T_i$ such that $|T_i|$ is maximum (for $i \in [1, 3]$). The $2.27^p + kn^3$ term comes from an algorithm using a reduction of MCT to 3-HITTING SET. Negative results have been obtained by Guillemot and Nicolas concerning the fixed-parameter tractability of MCT wrt the maximum degree $D$ of the input trees.

**Theorem 4.** *[7]*

1. MCT *is $W[1]$-hard with respect to $D$.*
2. MCT *can not be solved in $O(N^{o(2^{D/2})})$ time unless SNP $\subseteq$ SE, where $N$ denotes the input length,* i.e. *$N = O(kn)$.*

The MCT problem also admits a variant that deals with *supertrees*, *i.e.* trees having different (but overlapping) sets of leaves. The resulting problem is $W[2]$-hard with respect to $p$ [3].

## Approximation algorithms

The idea of locating and then eliminating successively all the conflicts between the input trees has also led to approximation algorithms for the *complement* version of the MCT problem, denoted CMCT. Let $L$ be the leaf set of each tree in an input collection $\mathcal{T}$, CMCT aims at selecting the smallest number of leaves $S \subseteq L$ such that the collection $\{T_i|(L-S) : T_i \in \mathcal{T}\}$ is compatible.

**Theorem 5.** *[6] Given a collection $\mathcal{T}$ of $k$ rooted trees on an $n$-leaf set $L$, there is a 3-approximation algorithm for* CMCT *that runs in $O(k^2n^2)$ time.*

The running time of this algorithm was later improved:

**Theorem 6.** *[1] There is an $O(kn+n^2)$ time 3-approximation algorithm for* CMCT *on a collection of $k$ rooted trees with $n$ leaves.*

Note also that working on rooted or unrooted trees does not change the achievable approximation threshold for CMCT [1].

# 3 APPLICATIONS

In bioinformatics, the MCT problem (and similarly MAST) is used to reach different practical goals. The first motivation is to measure the similarity of a set of trees. These trees can represent RNA secondary structures [10, 11] or estimates of a phylogeny inferred from different datasets composed of molecular sequences (*e.g.* genes) [13]. The gap between the size of a maximum compatible tree and the number of input leaves indicates the degree of disimilarity of the input trees. Concerning the phylogenetic applications, quite often some edges of the trees inferred from the datasets have been collapsed due to unsufficient statistical support, resulting in some higher-degree nodes in the trees considered by MCT. Each such node does not indicate a multi-speciation event but rather the uncertainty with respect to the branching pattern to be chosen for its child subtrees. In such a situation, the MCT problem is to be preferred to MAST, as it correctly handles high degree nodes, enabling them to be resolved according to branching information present in other input trees. As a result, more leaves are conserved in the output tree, hence a larger degree of similarity is detected between the input trees. Note also that a low similarity value between the input trees can be due to horizontal gene transfers. When these events are not too numerous, identifying species subject to such effects is done by first suspecting leaves discarded from a maximum compatible tree.

The shape of a maximum compatible tree, *i.e.* not just its size, also has an application in systematic biology to obtain a consensus of a set of phylogenies that are optimal for some tree-building criterion. For instance, the maximum parsimony and maximum likelihood criteria can provide several dozens (sometimes hundreds) of optimal or near-optimal trees. In practice, these trees are first grouped into islands of neighbouring trees, and a consensus tree is obtained for each island by resorting to a classical consensus tree method, *e.g.* the majority-rule or strict consensus. The trees representing the islands form a collection of which a consensus is then sought. However, consensus methods keeping all input leaves tend to create trees that lack of resolution. An alternative approach lies in proposing a representative tree that contains a largest possible subset of leaves on the position of which the trees of the collection agree. Again, MCT is more suited than MAST as the input trees can contain some high-degree nodes, with the same meaning as discussed above.

# 4 OPEN PROBLEMS

A direction for future work would be to examine the variant of MCT where some leaves are imposed in the output tree. This question arises when a biologist wants to ensure that the species central to his study are contained in the output tree. For MAST on two trees, this constrained variant of the problem was shown in a natural way to be of the same complexity as the regular version [4]. For MCT however, such a constraint can lead to several optimization problems that need to be sorted out. Another important work to be done is a set of experiments to measure the range of parameters for which the algorithms proposed to solve or approximate MCT are useful.

# 5 URL to CODE

A beta-version of a Perl program can be asked to the author of this entry.

## 6 CROSS REFERENCES

Maximum Agreement Subtree (of 2 Binary Trees) Entry 00178, Maximum Agreement Subtree (of 3 or More Trees) Entry 00177.

## 7 RECOMMENDED READING


[1] V. Berry, S. Guillemot, F. Nicolas, and C. Paul, *On the approximation of computing evolutionary trees*, in Proc. of the 11th Annual International Conference on Computing and Combinatorics (COCOON'05), L. Wang, ed., vol. 3595 of LNCS, Springer, 2005, pp. 115–125.

[2] V. Berry and F. Nicolas, *Improved parametrized complexity of the maximum agreement subtree and maximum compatible tree problems*, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3 (2006), pp. 289–302.

[3] ——, *Maximum agreement and compatible supertrees*, Journal of Discrete Algorithms, to appear (2006).

[4] V. Berry, Z. S. Peng, and H.-F. Ting, *From constrained to unconstrained maximum agreement subtree in linear time*, Algorithmica, to appear (2006).

[5] G. Ganapathy and T. J. Warnow, *Finding a maximum compatible tree for a bounded number of trees with bounded degree is solvable in polynomial time*, in Proc. of the 1st International Workshop on Algorithms in Bioinformatics (WABI'01), O. Gascuel and B. M. E. Moret, eds., 2001, pp. 156–163.

[6] ——, *Approximating the complement of the maximum compatible subset of leaves of k trees*, in Proc. of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX'02), 2002, pp. 122–134.

[7] S. Guillemot and F. Nicolas, *Solving the maximum agreement subtree and the maximum compatible tree problems on many bounded degree trees*, in Proc. of the 17th Combinatorial Pattern Matching Symposium (CPM'06), M. Lewenshtein and G. Valiente, eds., vol. 4009 of LNCS, Springer-Verlag, 2006, pp. 165–176.

[8] D. Gusfield, *Efficient algorithms for inferring evolutionary trees*, Networks, 21 (1991), pp. 19–28.

[9] A. M. Hamel and M. A. Steel, *Finding a maximum compatible tree is NP-hard for sequences and trees*, Applied Mathematics Letters, 9 (1996), pp. 55–59.

[10] J. Hein, T. Jiang, L. Wang, and K. Zhang, *On the complexity of comparing evolutionary trees*, Discrete Applied Mathematics, 71 (1996), pp. 153–169.

[11] T. Jiang, L. Wang, and K. Zhang, *Alignment of trees - an alternative to tree edit*, Theoretical Computer Science, 143 (1995), pp. 137–148.

[12] M. A. Steel and T. J. Warnow, *Kaikoura tree theorems: Computing the maximum agreement subtree*, Information Processing Letters, 48 (1993), pp. 77–82.

[13] D. Swofford, G. Olsen, P. Wadell, and D. Hillis, *Phylogenetic inference*, in Molecular systematics (2nd edition), D. Hillis, D. Moritz, and B. Mable, eds., Sunderland, USA, 1996, pp. 407–514.