Supplementary Material – Detecting Evolutionary Patterns of Cancers using Consensus Trees

Sarah Christensen[1] Juho Kim[2] Nicholas Chia[3,4] Oluwasanmi Koyejo[1]
Mohammed El-Kebir[1,*]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801
[2]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801
[3]Microbiome Program, Center for Individualized Medicine, Mayo Clinic, Rochester, MN 55905
[4]Division of Surgical Research, Department of Surgery, Mayo Clinic, Rochester, MN 55905

[*]Corresponding author: melkebir@illinois.edu

Contents

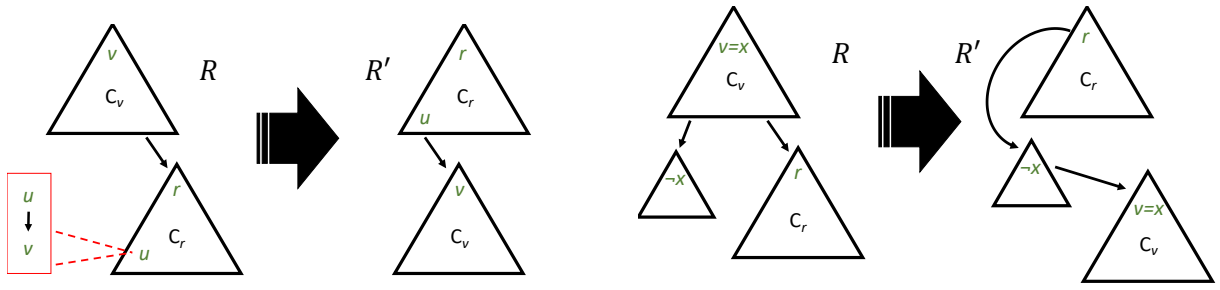


Fig S1: The two cases considered in Lemma 1 when the root vertex v of R does not equal r. (a) In the first case, subtree C_r of R contains an edge (u, v) that is present in at least one selected tree in Γ. (b) In the second case, no edge (u, v) of subtree C_r of R is present in at least one selected tree in Γ.

A NP-Hardness Proof

(Main Text) Theorem 1. MCCT is NP-hard even in the restricted case where (i) we seek a single consensus tree ($k = 1$), (ii) trees in $\mathcal{T}$ have the same vertex set Σ, and (iii) there are no mutation clusters.

We begin our proof of the theorem by making several observations about the structure of any optimal consensus tree, regardless of patient tree selection. The first observation we make is about the objective function.

Observation 1. Let (Γ, R) be composed tree selection $\Gamma = \{S_1, \ldots, S_n\}$ and consensus tree R. Then, (Γ, R) achieves minimum normalized parent-child distance $d_N(\Gamma, R)$ if and only if (Γ, R) achieves minimum unnormalized parent-child distance $d(\Gamma, R)$.

Proof. Since the vertex sets of the selected trees Γ and the consensus tree R are identical, we have

$$\begin{aligned} d_N(\Gamma, R) &= \sum_{i=1}^{n} \frac{|E(S_i) \triangle E(R)| + |V(S_i) \triangle V(R)|}{2|\Sigma|} \\ &= \frac{1}{2|\Sigma|} \sum_{i=1}^{n} |E(S_i) \triangle E(R)| = \frac{1}{2|\Sigma|} \sum_{i=1}^{n} d(S_i, R) = \frac{1}{2|\Sigma|} d(\Gamma, R). \end{aligned}$$

Thus, $d_N(\Gamma, R) \propto d(\Gamma, R)$.

□

Therefore, in the remainder of the proof, we will only consider unnormalized parent-child distances $d(\Gamma, R)$, which we refer to as simply 'parent-child distance'. Next, because r is the root across all input trees, we show in the following lemma that if r is not the root of the consensus tree R, we can construct a new consensus tree R' with a smaller parent-child distance to any selection from $\mathcal{T}(\phi)$. This leads to a contradiction on the optimality of R.

Lemma 1. Any optimal consensus tree R with respect to $\mathcal{T}(\phi)$ has r as the root vertex.

Proof. Let $\Gamma = \{S_1, \cdots, S_n\}$ denote the trees selected from $\mathcal{T}(\phi)$ minimizing the total parent-child distance to R. Let v be the root vertex of R. Suppose for a contradiction that $v \neq r$. We will show how to construct a tree R' from R that contradicts the optimality of R, i.e. $d(\Gamma, R) > d(\Gamma, R')$. First, we remove the incoming edge to r, disconnecting R into two components: C_v containing v and C_r containing r. Initially, we set R' equal to C_r. When adding the remaining vertices from C_v to R', there are two cases to consider (Fig. S1).

1. The first case is when C_r contains some vertex u such that (u, v) is an edge in at least one selected tree in Γ. In this case, we reattach C_v rooted at v as a child of u in R'. Removing the incoming edge to r decreases the total distance to Γ by n since r has no parent in any tree of Γ by construction. Adding the edge (u, v) can increase the distance by at most $n - 1$, since (u, v) appears in at least one tree in Γ. Thus, the overall distance decreases, i.e. $d(\Gamma, R) > d(\Gamma, R')$, contradicting the optimality of R.

2. The second case is when C_r does not contain any vertices that appear as a parent to v in a selected tree in Γ. By construction, each clause vertex has r as its parent in every selected tree in Γ. Since r is contained in C_r, v cannot correspond to a clause vertex. It must thus correspond to some variable vertex $v = x$ that is never picked as a child of r in Γ. This implies that (i) x is a child of $\neg x$ in at least two trees in Γ, (ii) $\neg x$ must be a child of r in the same two trees in Γ, and (iii) $\neg x$ must also be in C_v. Note that (i) and (ii) are because we assume every literal appears in at least two clauses, and (iii) is because C_r does not contain a parent of v. To construct R', we perform two additional operations. First, we remove the incoming edge to $\neg x$ in C_v, and we reattach the resulting subtree rooted at $\neg x$ as a child of r in R'. Second, we take the remaining component of C_v rooted at v and make it a child of $\neg x$ in R' (i.e., we add the edge $(\neg x, x)$).

   We now show that R' has a smaller total distance to Γ than R. Similar to the above case, removing the incoming edge to r decreases the total distance by n. Adding the edge $(\neg x, x)$ increases the distance by at most $n - 2$, since $(\neg x, x)$ appears in at least two trees in Γ. The final operation replaces the incoming edge to $\neg x$ with r. To see why the distance decrease, observe that $\neg x$ is a variable vertex. By construction the parent of $\neg x$ in any selected tree $S \in \Gamma$ is either (i) the negated variable vertex x, (ii) a clause variable, or (iii) the root r. By the premise, there is no selected tree $S \in \Gamma$ where x is the parent of $\neg x$. In addition, each clause variable can be at most once the parent of $\neg x$ among all selected trees Γ. Hence, r is the most frequent parent of $\neg x$ (occurring at least twice), and replacing the incoming edge to $\neg x$ with r also decreases the distance. Thus, we have contradicted the optimality of R implying that any optimal consensus tree must be rooted at r.

□

Since the edge (r, c_j) is in all input trees for all $j \in [n]$, these edges must also be present in any optimal consensus tree R. If these edges are not in the consensus tree, we again obtain a contradiction on the optimality of the consensus, as we show in the following lemma.

Lemma 2. Any optimal consensus tree R with respect to $\mathcal{T}(\phi)$ has the edge (r, c_j) for all $j \in [n]$.

Proof. Let $\Gamma = \{S_1, \cdots, S_n\}$ denote the trees selected from $\mathcal{T}(\phi)$ minimizing the total parent-child distance to R. Assume by way of contradiction that R does not contain the edge (r, c_j) for some j in $[n]$. We will construct a new consensus tree R' contradicting the optimality of R. The key fact to note is that r is the parent of c_j for every selected tree in $\Gamma = \{S_1, \cdots, S_n\}$ by construction. We consider two cases.

1. In the first case, the subtree rooted at c_j does not contain r. Let R' be obtained from R by regrafting the subtree rooted at c_j as a child of r. The parent-child distance will decrease by 2 for each selected tree in Γ, as the incoming edge to c_j now matches for all pairs of trees. Hence, $d(\Gamma, R) > d(\Gamma, R')$, contradicting the optimality of R.

2. In the second case, the subtree rooted at c_j contains r in R. This implies that r is not the root of R. By Lemma 1, this contradicts the optimality of R.

We get a contradiction in both cases. Thus, a solution of MCCT on $\mathcal{T}(\phi)$ has edges $\{(r, c_j)\}$ for all $j \in [n]$. □

We now proof a lower bound on the total parent-child distance of an optimal consensus tree R of the input trees $\mathcal{T}$. At a high level, this lower bound comes from the fact that edges of the form (c_i, x), for some clause vertex c_i and variable vertex x, only appear in the ith patient's set of input trees. There are $2m - 6$ such edges of this form in any selected patient tree $S_i \in \mathcal{T}_i$. We use this to identify $2(2m - 6)$ edges that must be in the symmetric difference between R and S_i. After doing this over all n patients and taking care to avoid double counting, we establish the lower bound of $2n(2m - 6)$.

Lemma 3. An optimal consensus tree R with respect to $\mathcal{T}(\phi)$ has a parent-child distance of at least $2n(2m-6)$. Moreover, a consensus tree R achieving this lower bound cannot have edges of the form (c_i, x) or $(c_i, \neg x)$ for some clause c_i and variable x.

Proof. Let R be a consensus tree and $\Gamma = \{S_1, \cdots, S_n\}$ be a set of trees selected from $\mathcal{T}(\phi)$ minimizing the total parent-child distance $d(\Gamma, R)$. Let Δ be the multi-set composed of the symmetric differences $(E(S_i) \setminus E(R)) \cup (E(R) \setminus E(S_i))$ where $i \in [n]$. We claim that $|\Delta| \geq 2n(2m - 6)$, which implies that $d(\Gamma, R) \geq 2n(2m - 6)$. We will prove this constructively using the following algorithm.

1. $\Delta \leftarrow \emptyset$
2. **for** $i \leftarrow [n]$ **do**
3. Let X_i be the set of literals corresponding to variables absent in c_i
4. **for** $x \leftarrow X_i$ **do**
5. **if**$(c_i, x) \notin E(R)$ **then**
6. Let (v, x) be the unique incoming edge to x in R
7. $\Delta \leftarrow \Delta \cup \{(c_i, x) \in E(S_i), (v, x) \in E(R)\}$
8. **else**
9. Let c_j be a clause containing x and let (v, x) be the unique incoming edge to x in S_j
10. $\Delta \leftarrow \Delta \cup \{(v, x) \in E(S_j), (c_i, x) \in E(R)\}$

We claim that at iteration i, we have identified $2i(2m - 6)$ edges of Δ, which we prove by induction on i.

- *Base case* $i = 1$: Consider $S_1 \in \Gamma$ corresponding to clause c_1. Let X_1 be the set of literals corresponding to variables that are absent in c_1. Since the literals from each clause in ϕ come from three distinct variables, there are a total of $|X_1| = 2m - 6$ literals corresponding to variables absent in c_1. Thus, there are a total of $|X| = 2m - 6$ edges (c_1, x) in tree S_1 where $x \in X_1$.

  For each such edge (c_1, x) in S_1, there are two cases. If (c_1, x) is not in R, then (c_1, x) in S_1 increases the symmetric difference by 1. Furthermore, since x is not the root of R (by Lemma 1), the incoming edge to x in R is missing from S_1 (i.e., (v, x) for some $v \in V$). Altogether, this increased the symmetric difference by 2.

  Now consider the case where (c_1, x) is in R. By construction, (c_1, x) is not in any other input tree in $\mathcal{T}(\phi) \setminus \{\mathcal{T}_1(\phi)\}$ and is thus also absent from $\{S_2, \ldots, S_n\}$. However, we must charge this edge carefully in order to not double count edges across selected trees in future steps. By our restrictions on ϕ, x appears in some clause, c_j, for some j in $[n]$. By construction, the corresponding selected tree S_j must then contain the edge (v, x) where $v \in \{r, \neg x\}$. Either way, x has a different parent $v \neq c_1$ in S_j compared to R and we have identified two edges, (c_1, x) and (v, x) of Δ.

Repeating this process for all $2m-6$ edges in S_1 where c_1 is the parent, we find $2(2m-6)$ distinct edges to add to the symmetric difference.

- *Inductive step* $i > 1$: By the inductive hypothesis, we assume we are able to identify $2(i-1)(2m-6)$ distinct edges of Δ. Now consider the ith selected tree in our ordering, S_i. We claim that we can identify an additional $2(2m-6)$ distinct edges of Δ. Let X_i be the set of literals corresponding to variables that are absent in c_i. As before, the selected tree S_i contains $2m-6$ edges (c_i, x) where $x \in X_i$.

  For each such edge (c_i, x) in S_i, we again distinguish two cases. If (c_i, x) is not in R, then R and S_i must have different parents for x. Indeed, as described in the base case, (c_i, x) is present in S_i but not in R and conversely the incoming edge (v, x) to x in R is also missing from S_i (which exists, as x cannot be the root of R by Lemma 1). Hence, the edge (c_i, x) of S_i and the (v, x) of R are edges of Δ. We now need to show that the edge (c_i, x) of S_i was not added in a previous iteration to Δ. To see why this is not the case, observe that only two types of edges from Γ were previously added. The first type are edges (c_j, x) that were added in step $j < i$. The second type are edges (v, x) of a tree S_l where $v \in \{r, \neg x\}$, corresponding to a variable x that is *present* in some clause c_l. Both cases do not apply, as $j < i$ and $c_i \notin \{r, \neg x\}$. Hence, these two edges of Δ were not previously considered.

  The second case is when (c_i, x) is in R. Let c_l be a clause containing x (which must exist by definition of ϕ). By construction, the corresponding selected tree S_l contains the edge (v, x) where $v \in \{r, \neg x\}$. Since $c_i \neq v$, clearly the edge (c_i, x) of R and the edge (v, x) of S_l are present in Δ. We claim that the edge (v, x) of S_l was not added to Δ in a previous iteration $j < i$. Inspection of the algorithm reveals that this edge (v, x) can only be added in a previous iteration j if (c_j, x) is an edge of R. However, by our premise, R already contains the edge (c_i, x). Thus, since R is a tree, there is no edge (c_j, x) in R where $c_j \neq c_i$. Hence, the edge (v, x) of Δ was not previously considered.

  Repeating this process for all $2(2m-6)$ edges in S_i where c_i is the parent, we find an additional $2(2m-6)$ distinct edges to add to the symmetric difference. Combining this total with the inductive hypothesis we have identified $2i(2m-6)$ distinct edges of Δ upon completion of iteration i.

Thus, when the algorithm terminates at iteration n, we have identified $2n(2m-6)$ distinct edges of Δ. To prove the final part of this lemma, we note that each literal x appears in at least two clauses by our assumption on ϕ. In the case where (c_i, x) is in R, we can therefore find two edges to add to the symmetric difference for *every* selected tree S_j corresponding to a clause c_j containing x (i.e., at least 4 edges will be added). In this case, R fails to achieve the lower bound of $2n(2m-6)$. □

The following lemma again follows from a proof by contradiction on the optimality of R if this were not the case.

Lemma 4. Let consensus tree R and selected trees $\Gamma = \{S_1, \ldots, S_n\}$ be an optimal solution to MCCT instance $\mathcal{T}(\phi)$. If $d(\Gamma, R) = 2n(2m-6)$ then either $\{(r, x), (x, \neg x)\} \subseteq E(R)$ or $\{(r, \neg x), (\neg x, x)\} \subseteq E(R)$ for all variables x. Moreover, whichever set appears in $E(R)$ must also occur in every tree of Γ_x, where Γ_x denotes the subset of selected trees Γ corresponding to clauses containing x or $\neg x$.

Proof. Let consensus tree R and selected trees $\Gamma = \{S_1, \ldots, S_n\}$ be an optimal solution to MCCT instance $\mathcal{T}(\phi)$ with total distance $d(\Gamma, R) = 2n(2m-6)$. Suppose by way of contradiction that there exists a variable x such that neither $\{(r, x), (x, \neg x)\}$ nor $\{(r, \neg x), (\neg x, x)\}$ appear in $E(R)$. We consider three cases for the arrangement of x and $\neg x$ in R.

(i) Vertex x or $\neg x$ is the root of R:

This case contradicts the optimality of R by Lemma 1.

(ii) Vertex x or $\neg x$ has a different variable vertex as a parent:

Let $y \neq \neg x$ be a variable vertex that is parent of x (the case for $\neg x$ is symmetric). As this arrangement never appears in any input tree in $\mathcal{T}(\phi)$, we have the edge (y, x) of R incurs a cost of 2 in each selected tree in Γ. Thus, it is straightforward to construct a tree R' with a lower distance to Γ. That is, consider a selected tree $S \in \Gamma_x$. If S contains the edge (r, x) then move x to be the child of r in R'. Otherwise, if S contains the edge $(\neg x, x)$ then move x to be the child of $\neg x$ in R'. In both cases, the total distance $d(\Gamma, R')$ is strictly smaller than the original distance $d(\Gamma, R)$ (by at least a value of 2), leading to a contradiction.

(iii) Vertex x or $\neg x$ has a clause variable as a parent:

This contradicts R achieving the lower bound distance $d(\Gamma, R) = 2n(2m - 6)$ by Lemma 3.

(iv) Both x and $\neg x$ have r as a parent:

No clause variable c_i for $i \in [n]$ has a child in R by Lemma 3. We can obtain the lower bound distance of $2n(2m - 6)$ by counting edges of the form $(c_i, x) \subset E(S_i)$ across all S_i that must be in the multi-set Δ composed of symmetric differences. For each such edge (c_i, x), a counterpart edge $(p, x) \subset E(R)$ for some $p \in V$ must also be in Δ. We now need to find one more edge in Δ to obtain a contradiction. By construction, Γ_x cannot be empty and each tree in Γ_x must either contain $\{(r, x), (x, \neg x)\}$ or $\{(r, \neg x), (\neg x, x)\}$. WLOG assume $S \in \Gamma_x$ contains $\{(r, x), (x, \neg x)\}$. Then, $(x, \neg x) \subset E(S)$ must also be in δ, contradicting the fact that R achieves the lower bound.

Thus, either $\{(r, x), (x, \neg x)\} \subseteq E(R)$ or $\{(r, \neg x), (\neg x, x)\} \subseteq E(R)$ for all variables x. To prove the final point, WLOG assume $\{(r, x), (x, \neg x)\} \subseteq E(R)$; if there exists a tree in $S \in \Gamma_x$ such that $\{(r, \neg x), (\neg x, x)\} \subseteq E(S)$, both of these edges must also exist in Δ implying that R does not achieve the lower bound.

□

In one direction, we now show that given an optimal consensus tree with parent-child distance $2n(2m - 6)$, we can read off the satisfying assignment by looking at the children of the root vertex r in R. In the other direction, we use a satisfying assignment to identify which tree we should select for each patient (i.e. the one corresponding to a satisfying assignment) and build the consensus tree R, where the satisfied literals hang off of the root.

Lemma 5. A Boolean formula ϕ meeting our three restrictions is satisfiable if and only if $\mathcal{T}(\phi)$ has an optimal single consensus tree with parent-child distance $2n(2m - 6)$.

Proof. ($\Rightarrow$) Let ϕ be satisfiable. We will directly construct a solution to the corresponding MCCT problem achieving the lower bound. Let θ be a satisfying assignment. Consider an arbitrary clause c_i containing variables x_1, x_2, x_3. We select tree $S_i \in \mathcal{T}_i$ such that $(r, x_j) \in E(S_i)$ if $\theta(x_j) = 1$ and $(r, \neg x_j) \in E(S_i)$ if $\theta(x_j) = 0$. Note that such a tree must exist in $\mathcal{T}_i$ by construction. We construct a consensus tree R with root r and edges (r, c_i) for all $i \in [n]$. We then add edges $\{(r, x)(x, \neg x\}$ if $\theta(x_j) = 0$ or edges $\{(r, \neg x)(\neg x, x\}$ if $\theta(x_j) = 1$. Finally, observe that $d(S_i, R)$ is equal to $2(2m - 6)$, due to edges of the form $(c_i, x) \in E(S_i)$ and the corresponding edge $(v, x) \in E(R)$ where $v \in \{r, \neg x\}$. Since i was arbitrary, we constructed consensus tree R with distance $2n(2m - 6)$ across all selected trees. By Lemma 3, R must be optimal.

($\Leftarrow$) Let $\mathcal{T}(\phi)$ have an optimal consensus tree R and tree selection Γ with distance $d(\Gamma, R) = 2n(2m - 6)$. By Lemma 4, R must have either the edge (r, x) or $(r, \neg x)$ for all variables x (but not both). Consider the assignment θ which sets a variable x equal to 1 if literal x is a child of the root and equal to 0 if literal $\neg x$ is a child of the root in R. We claim θ is a satisfying assignment for ϕ. Let c_i be an arbitrary clause in ϕ, and let $S_i \in \mathcal{T}_i$ be the tree selected by MCCT. For each variable x in clause c_i, either $(r, x) \in E(S_i)$

if $\theta(x) = 1$ or $(r, \neg x) \in E(S_i)$ if $\theta(x) = 0$ by Lemma 4. By construction of trees in $\mathcal{T}_i$, this implies that $\theta(x_j) = \gamma(y_i, j)$ for $j \in [3]$; thus, θ corresponds to a satisfying assignment for clause c_i. Since c_i was arbitrary, all clauses in ϕ must be satisfied. □

This then concludes the proof of (Main Text) Theorem 1 as we have now shown that MCCT is hard even for this special case where $k = 1$ on identical patient mutation sets with no mutation clusters by a polynomial reduction from 3-SAT; thus, MCCT is NP-hard in general.

B Dynamic programming algorithm for expanding mutation clusters

Here we describe the dynamic programming (DP) algorithm we use to solve the OCE problem. Recall that in this problem, we are given a tree R with no mutation clusters and a tree T with at least one mutation cluster. We wish to find a tree T' such that (i) T' is an expansion of T, and (ii) T' minimizes the normalized parent-child distance to R out of all tree expansion of T.

In the following recursion of our DP, each subproblem is an expansion of a subtree of T. For each subtree, we look at all possible start and end mutations for the expansion of the root mutation cluster; intuitively, these are the mutations that interact with mutations outside of the cluster. Let T_C be the subtree of T rooted at the vertex corresponding to mutation cluster C and let $\mu(C)$ be the set of mutations in C. Given R and T as defined above, we define the function $f(C, s, t)$ to be the maximum number of matching pairs of edges between R and any expansion of the subtree T_C, such that the mutation cluster $C \in V(T)$ is expanded into a path starting with mutation s and ending with mutation t.

$$f(C, s, t) = \begin{cases} -\infty, & \text{if } |\mu(C)| > 1 \text{ and } s = t, \\ 0, & \text{if } |\mu(C)| = 1 \text{ and } C \text{ is a leaf}, \\ g(C, s, t) + h(C, s, t), & \text{otherwise}, \end{cases}$$

where we have

$$h(C, s, t) = \sum_{W \in \delta(C)} \max_{s', t' \in \mu(W)} \left\{ \mathbb{1}((t, s') \in E(R)) + f(W, s', t') \right\},$$

which recursively finds the best scoring expansion for the children $\delta(C)$ of C given t, adding an additional match if there is an edge between t and the expansion of a child in $E(R)$. In the recurrence, we have $g(C, s, t)$, which is defined as the maximum number of matching pairs of edges between R and any expansion of the mutation cluster C starting with mutation s and ending with mutation t. If $s = t$, this is defined to be zero. For now, assume we have oracle access to this value. We will give an explicit algorithm for calculating $g(C, s, t)$ later in this section.

Theorem 2. Given a rooted tree R with no mutation clusters and a rooted tree T with at least one mutation cluster, taking $OCE(T, R) = \max_{s,t \in \mu(r(T))} f(r(T), s, t)$ finds the optimal value for the OCE problem.

Proof. We prove this theorem by induction on $\ell \in [|V(T)|]$, where ℓ denotes the index of a vertex within a topological ordering of $V(T)$ such that each child vertex comes prior to its parent.

Base case: When $\ell = 0$, we have that the vertex C_0 must be a leaf. Let $s, t \in \mu(C_0)$ be arbitrary. There are three cases to consider:

1. If C_0 is not a mutation cluster (i.e., $|\mu(C_0)| = 1$, $s = t$), then there are no edges in the expansion of T_{C_0} since this is a leaf. Hence, $f(C_0, s, t) = 0$ as claimed.

2. Else if C_0 is a mutation cluster and $s = t$, then by definition of f the mutation s needs to be repeated twice, once at the start and once at the end of the expansion. Since this degenerate case is not allowed, f correctly returns $-\infty$ so it is never selected. Indeed, any pair of distinct mutations from C_0 will achieve a better score.

3. Finally, if C_0 is a mutation cluster and $s \neq t$, $f(C, s, t) = g(C_0, s, t)$ since $\delta(C_0) = \emptyset$, i.e., C_0 has no children. This is again correct by the definition of $g(C_0, s, t)$.

Inductive step: Assume the recursion is correct for all subtrees up to the one rooted at $C_{\ell-1}$. We now consider the subtree rooted at C_ℓ. There are again three cases to consider:

1. If C_ℓ is a leaf, then the correctness follows from the same analysis as in the base case.

2. Else if C_ℓ is a mutation cluster and $s = t$, then this degenerate case is not allowed. As was shown in the base case, f correctly returns $-\infty$ so $s = t$ is never selected.

3. Finally, if C_ℓ is a mutation cluster and $s \neq t$, the score of an expansion of T_{C_ℓ} is simply the sum of (i) the number of edges from an expansion of C_ℓ starting with s and ending with t that exist in $E(R)$, (ii) the number of edges between t and the root of the expansion of each child in $E(R)$, and (iii) the score of the expansion of the subtree at each child. We see that $g(\cdot, \cdot, \cdot)$ maximizes (i) while $h(\cdot, \cdot, \cdot)$ jointly maximizes (ii) and (iii).

It then follows that $\max_{s,t \in \mu(r(T))} f(r(T), s, t)$ is the maximum number of matching pairs of edges between R and any expansion of T, since we exhaust all starting and ending points in the expansion of the root. Let T' be an expansion of T that achieves this score. This implies that T' maximizes $|E(T') \cap E(R)|$, which in turn implies that it minimizes the parent-child distance $d(T', R)$ out of all expansions of T. Note that since all expansions of T have the same number of edges, T' also minimizes $d_N(T', R)$ and is an optimal solution to the OCE problem. □

Next, we show how we can efficiently calculate $g(C, s, t)$ with respect to tree R. The pseudocode is given in Algorithm 1. The intuitive idea is we wish to identify all edges in $E(R)$ that can be preserved when expanding mutation cluster C. An upper bound on this number is of course the edges in $E(R)$ with both endpoints in $\mu(C)$. We call this restricted subgraph $R_{\mu(C)}$.

Definition 1. The graph R *restricted* to vertex set Σ, denoted R_Σ, is a directed graph on vertex set Σ with edges $\{(u, v) \in E(R) | u \in \Sigma, v \in \Sigma\}$.

However, it is not always possible to maintain all of these edges since R is a tree, and we expand C into a path. Therefore, we first identify the connected components of $R_{\mu(C)}$. To decompose these components into paths we perform the following operations:

1. If there is a path from s and t in some component, we carefully select an edge to break along this path since all the mutations will eventually need to be added somewhere between s and t. In particular, we break the edge with the highest degree parent along the path.

2. We also break any incoming edge to s or outgoing edge from t since these must be the start and endpoints of the expansion, respectively.

3. We then break along edges where a parent has more than one child to remove any remaining branches.

Finally, we stitch the resulting paths back together into one long path $\Pi(C)$, which we treat as the expansion of C. We ensure that this expansion starts with s and ends with t, but otherwise the ordering does not matter.

Algorithm 1: Single Cluster Expansion

Input: A rooted tree R with no mutation clusters, a rooted tree T with at least one mutation cluster, a mutation cluster $C \in V(T)$, and mutations $s, t \in \mu(C)$.

Output: The maximum number of edges shared with $E(R)$ in an expansion of C starting with s and ending with t.

if $|\mu(C)| > 1$ and $s = t$ **then**
  Raise Error
if $|\mu(C)| = 1$ **then**
  return 0
$\Theta \leftarrow$ Construct $R_{\mu(C)}$. Store resulting connected components.
if $\exists$ an edge (u, s) within a component $\ell \in \Theta$ **then**
  Split ℓ into two components by removing (u, s). Update Θ.
if $\exists$ an edge (t, v) within a component $\ell \in \Theta$ **then**
  Split ℓ into two components by removing (t, v). Update Θ.
if $\exists$ a path p from s to t within a component $\ell \in \Theta$ **then**
  Find vertex u on p with highest degree.
  Split ℓ into two components by removing the unique edge (u, v) such that v is on path p. Update Θ.
while $\exists \ell \in \Theta$, $v \in \ell$ such that v has more than one child **do**
  Remove edges to all but one child of v. Update Θ.
$\Pi(C) \leftarrow \ell \in \Theta$ such that $s \in \ell$.
for $\ell \in \Theta$ such that $s, t \notin \ell$ **do**
  Append component ℓ to path $\Pi(C)$ (i.e., $\Pi(C) \leftarrow \Pi(C) + \ell$).
$\Pi(C) \leftarrow \ell \in \Theta$ such that $t \in \ell$.
return $|E(\Pi(C)) \cap E(R)|$

Theorem 3. Algorithm 1 finds the maximum number of matching pairs of edges between R and any expansion of the mutation cluster C starting with mutation s and ending with mutation t.

We prove this theorem by first establishing an upper bound on this distance and then showing our algorithm achieves the upper bound.

Lemma 6. The maximum number of matching pairs of edges between R and any expansion $\Pi(C)$ of the mutation cluster C starting with mutation s and ending with mutation t is less than or equal to:

$$|E(R_{\mu(C)})| - \mathbb{1}(s) - \mathbb{1}(t) - \mathbb{1}(s,t) - \sum_{v \in V(R_{\mu(C)})} (|\delta(v)| - 1)$$

where $\mathbb{1}(s)$ indicates if s has a parent of outdegree 1 in $R_{\mu(C)}$, $\mathbb{1}(t)$ indicates if t has a child in $R_{\mu(C)}$, and $\mathbb{1}(s,t)$ indicates if there is a path from s to t in $R_{\mu(C)}$ without a vertex having more than one child.

A sketch for the proof of this lemma is the following. We can only keep one edge per parent vertex in $R_{\mu(C)}$ when we build the expansion path. We also must break any incoming edge to s and any outgoing edge from t. Finally, we also need to break an edge on the path from s to t to ensure that s is the starting vertex and t is the ending vertex, with all other vertices spliced between them. We want to make sure we do not double count broken edges, which results in the use of indicator variables.

Lemma 7. Algorithm 1 finds

$$|E(R_{\mu(C)}| - \mathbb{1}(s) - \mathbb{1}(t) - \mathbb{1}(s,t) - \sum_{v \in V(R_{\mu(C)})} (|\delta(v)| - 1)$$

matching pairs of edges between R and any expansion of the mutation cluster C starting with mutation s and ending with mutation t.

The proof of this lemma follows by counting the number of edges broken by Algorithm 1. Finally, we observe that the running time of Algorithm 1 is $O(|\Sigma|^5)$. To see this, note that there are $|V(T)| = O(|\Sigma|)$ possible roots $C \in V(T)$, each with $|\mu(C)|^2 = O(|\Sigma|^2)$ start and endpoints implying there are $O(|\Sigma|^3)$ distinct subproblem of $f(\cdot,\cdot,\cdot)$. Each subproblem then requires $O(|\Sigma|)$ time to compute $g(\cdot,\cdot,\cdot)$ and $h(\cdot,\cdot,\cdot)$. The former requires $O(\Sigma)$ time since it is comprised of a constant number of graph traversals and the latter requires $O(\Sigma^2)$ time when we carefully account for the number of times a child is recursed on:

$$\begin{aligned}
&O\left(\sum_{C \in V(T)} \sum_{s,t \in \mu(C)} \left[|\Sigma| + \sum_{W \in \delta(C)} \sum_{s',t' \in \mu(W)} 1\right]\right) \\
&= O\left(|\Sigma|^4 + \sum_{C \in V(T)} \sum_{s,t \in \mu(C)} \sum_{W \in \delta(C)} \sum_{s',t' \in \mu(W)} 1\right) \\
&= O\left(|\Sigma|^4 + \sum_{C \in V(T)} \sum_{W \in \delta(C)} \sum_{s,t \in \mu(C)} \Sigma^2\right) \\
&= O\left(|\Sigma|^4 + \sum_{C \in V(T)} \sum_{W \in \delta(C)} \Sigma^2 \cdot \Sigma^2\right) \\
&= O(\Sigma^5)
\end{aligned}$$

C Results

We ran HINTRA (Khakabimamaghani *et al.*, 2019) using the following arguments:

```
$ ./Hintra-Lin -u <INPUT_FILE> <no_samples> <no_genes> 0.1 50 10
```

where 0.1 is the default of the discretization parameter, 50 is the default value for the number of EM restarts and 10 is the number of threads. We ran REVOLVER (Caravagna *et al.*, 2018) using the following default of the following R function:

```
revolver_fit(x, initial.solution = 1, max.iterations = 10, n = 10)
```

We have the following supplemental results figures.

- Fig. S2 shows simulation results for $|\Sigma| = \ell = 5$.
- Fig. S3 shows simulation results for $|\Sigma| = 12$ and $\ell = 7$.
- Fig. S4 shows simulation results for $|\Sigma| = \ell = 12$.
- Fig. S5 shows performance of RECAP with varying number of restarts in simulations.

Algorithm 2: Generalized Coordinate Ascent Heuristic

Input: A collection $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ of patients' tree sets and number $k > 0$ of clusters

Output: Selection of trees $\{S_1, \cdots, S_n\}$, consensus trees $\{R_1, \cdots, R_k\}$, and clustering σ with smallest criterion score found.

$\mathcal{T}' \leftarrow$ augment the tree for each patient i from $\mathcal{T}_i$ to span all mutations plut $\perp$

$\{S_1, \cdots, S_n\} \leftarrow$ random tree selection for each patient i from $\mathcal{T}'_i$

$\sigma \leftarrow$ random surjective cluster mapping from $[n] \to [k]$

$\{R_1, \cdots, R_k\} \leftarrow$ Compute initial consensus tree for each cluster j by running SCT algorithm on the set $\{S_i | \sigma(i) = j\}$.

$\Delta \leftarrow \infty, L \leftarrow \sum_{i=1}^{n} OCE(S_i, R_{\sigma(i)})$

while $\Delta > 0$ **do**

  for $j \leftarrow 1$ **to** k **do**

    $R_j \leftarrow$ Update consensus tree for cluster j by running SCT algorithm on the set $\{S_i | \sigma(i) = j\}$.

  for $i \leftarrow 1$ **to** n **do**

    $S_i, \sigma(i) \leftarrow$ Update selected tree and cluster for patient i by directly computing $\text{argmin}_{T \in \mathcal{T}'_i, j \in [k]} OCE(T, R_j)$

  $\Delta \leftarrow L - \sum_{i=1}^{n} OCE(S_i, R_{\sigma(i)})$

  $L \leftarrow \sum_{i=1}^{n} OCE(S_i, R_{\sigma(i)})$

Remove $\perp$ and all its descendants for all trees in the selected set $\{S_1, \cdots, S_n\}$ and consensus set $\{R_1, \cdots, R_k\}$

return $(\{S_1, \cdots, S_n\}, \{R_1, \cdots, R_k\}, \sigma)$

- Fig. S6 shows the first five consensus trees identified by RECAP in a non-small cell lung cancer cohort.
- Fig. S7 shows the second five consensus trees identified by RECAP in a non-small cell lung cancer cohort.
- Fig. S8 shows the first four consensus trees identified by RECAP in a breast cancer cohort.
- Fig. S9 shows the second four consensus trees identified by RECAP in a breast cancer cohort.

References

Caravagna, G., Giarratano, Y., Ramazzotti, D., Tomlinson, I. P. M., Graham, T. A., Sanguinetti, G., and Sottoriva, A. (2018). Detecting repeated cancer evolution from multi-region tumor sequencing data. *Nature Methods*, **15**, 707–714.

Jamal-Hanjani, M., Wilson, G. A., McGranahan, N., Birkbak, N. J., Watkins, T. B., Veeriah, S., Shafi, S., Johnson, D. H., Mitter, R., Rosenthal, R., Salm, M., Horswell, S., Escudero, M., Matthews, N., Rowan, A., Chambers, T., Moore, D. A., Turajlic, S., Xu, H., Lee, S.-M., Forster, M. D., Ahmad, T., Hiley, C. T., Abbosh, C., Falzon, M., Borg, E., Marafioti, T., Lawrence, D., Hayward, M., Kolvekar, S., Panagiotopoulos, N., Janes, S. M., Thakrar, R., Ahmed, A., Blackhall, F., Summers, Y., Shah, R., Joseph, L., Quinn, A. M., Crosbie, P. A., Naidu, B., Middleton, G., Langman, G., Trotter, S., Nicolson, M., Remmen, H., Kerr, K., Chetty, M., Gomersall, L., Fennell, D. A., Nakas, A., Rathinam, S., Anand, G., Khan, S., Russell, P., Ezhil, V., Ismail, B., Irvin-Sellers, M., Prakash, V., Lester, J. F., Kornaszewska, M., Attanoos, R., Adams, H., Davies, H., Dentro, S., Taniere, P., OSullivan, B., Lowe, H. L., Hartley, J. A., Iles, N., Bell, H., Ngai, Y., Shaw, J. A., Herrero, J., Szallasi, Z., Schwarz, R. F., Stewart, A., Quezada, S. A., Le Quesne, J., Van Loo, P., Dive, C., Hackshaw, A., and Swanton, C. (2017). Tracking the evolution of nonsmall-cell lung cancer. *New England Journal of Medicine*, **376**(22), 2109–2121.

Khakabimamaghani, S., Malikic, S., Tang, J., Ding, D., Morin, R., Chindelevitch, L., and Ester, M. (2019). Collaborative intra-tumor heterogeneity detection. *Bioinformatics*, **35**(14), i379–i388.

Razavi, P., Chang, M. T., Xu, G., Bandlamudi, C., Ross, D. S., Vasan, N., Cai, Y., Bielski, C. M., Donoghue, M. T., Jonsson, P., *et al.* (2018). The genomic landscape of endocrine-resistant advanced breast cancers. *Cancer cell*, **34**(3), 427–438.

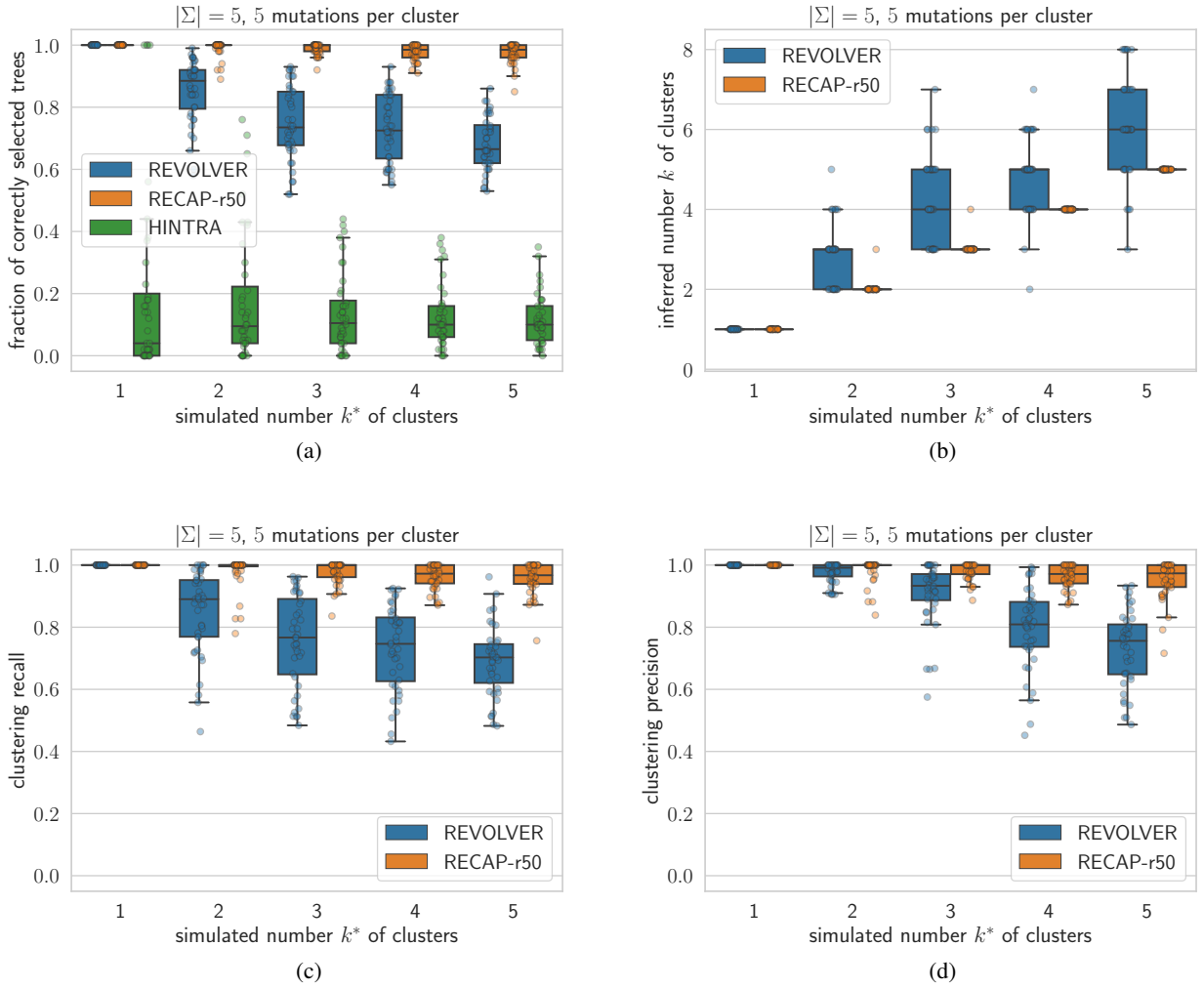


Fig S2: **Simulations show that RECAP accurately solves the MCCT problem for simulations with $|\Sigma| = 5$ total mutations and 5 mutations in each cluster.** (a) The fraction of patients with correctly inferred trees by each method. (b) The number k of patient clusters inferred by each method. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. No results are shown in (b)-(d) for HINTRA, as this method does not infer patient clusters.

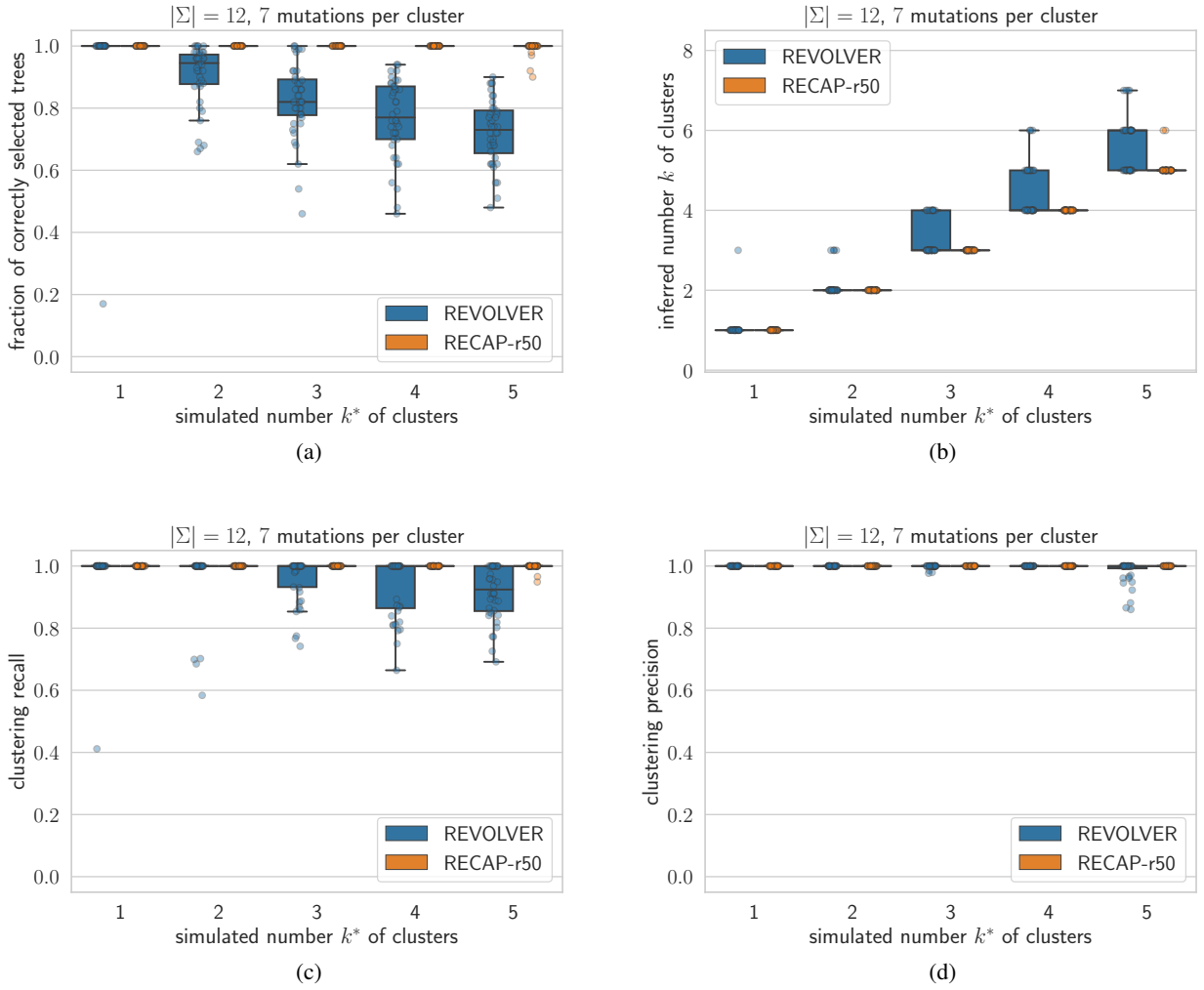


Fig S3: **Simulations show that RECAP accurately solves the MCCT problem for simulations with $|\Sigma| = 12$ total mutations and 7 mutations in each cluster.** (a) The fraction of patients with correctly inferred trees by each method. (b) The number k of patient clusters inferred by each method. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. No results are shown in (a)-(d) for HINTRA, as this method does not infer patient clusters and does not scale to the simulated number of mutations.

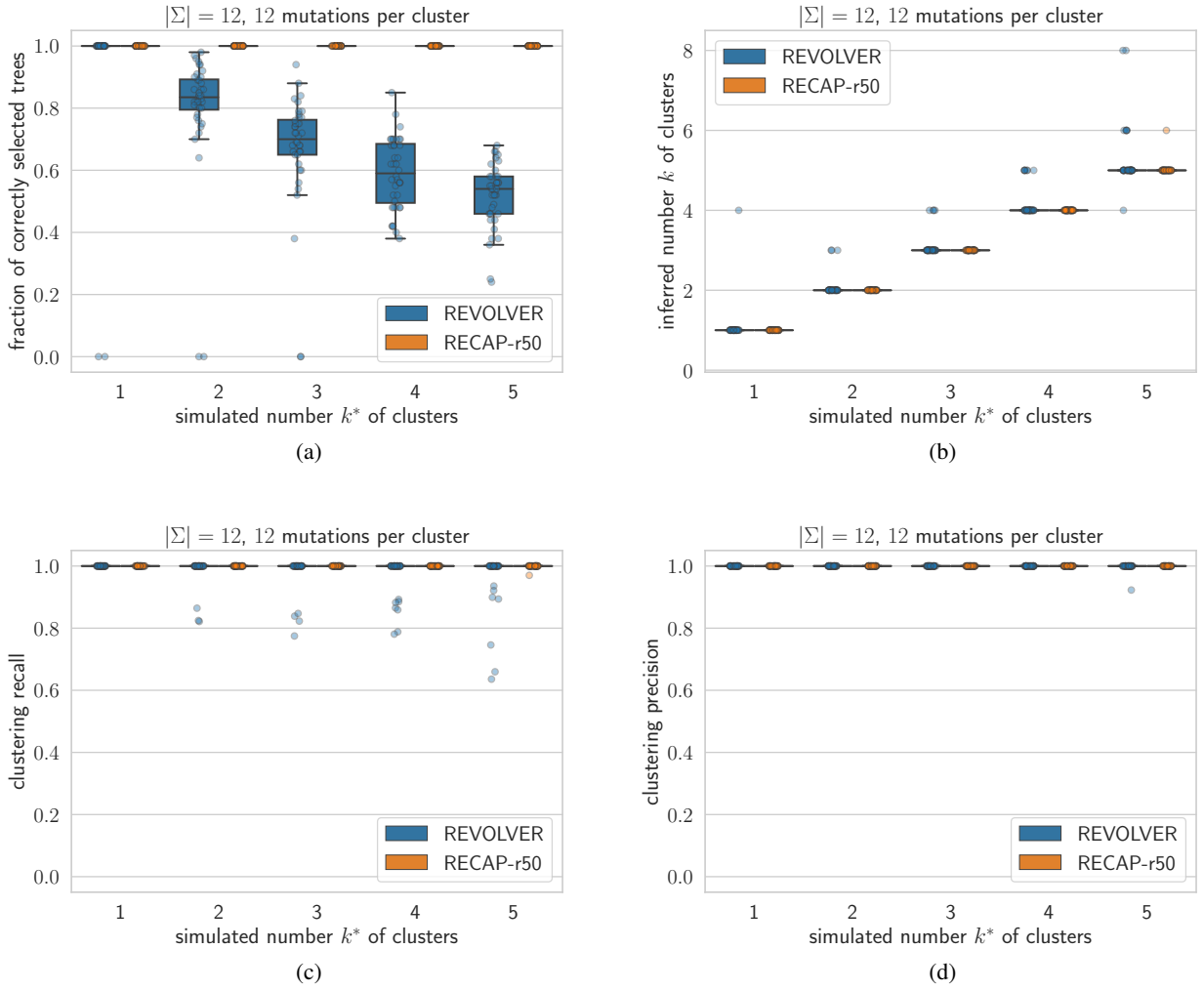


Fig S4: **Simulations show that RECAP accurately solves the MCCT problem for simulations with $|\Sigma| = 12$ total mutations and 12 mutations in each cluster.** (a) The fraction of patients with correctly inferred trees by each method. (b) The number k of patient clusters inferred by each method. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. No results are shown in (a)-(d) for HINTRA, as this method does not infer patient clusters and does not scale to the simulated number of mutations.

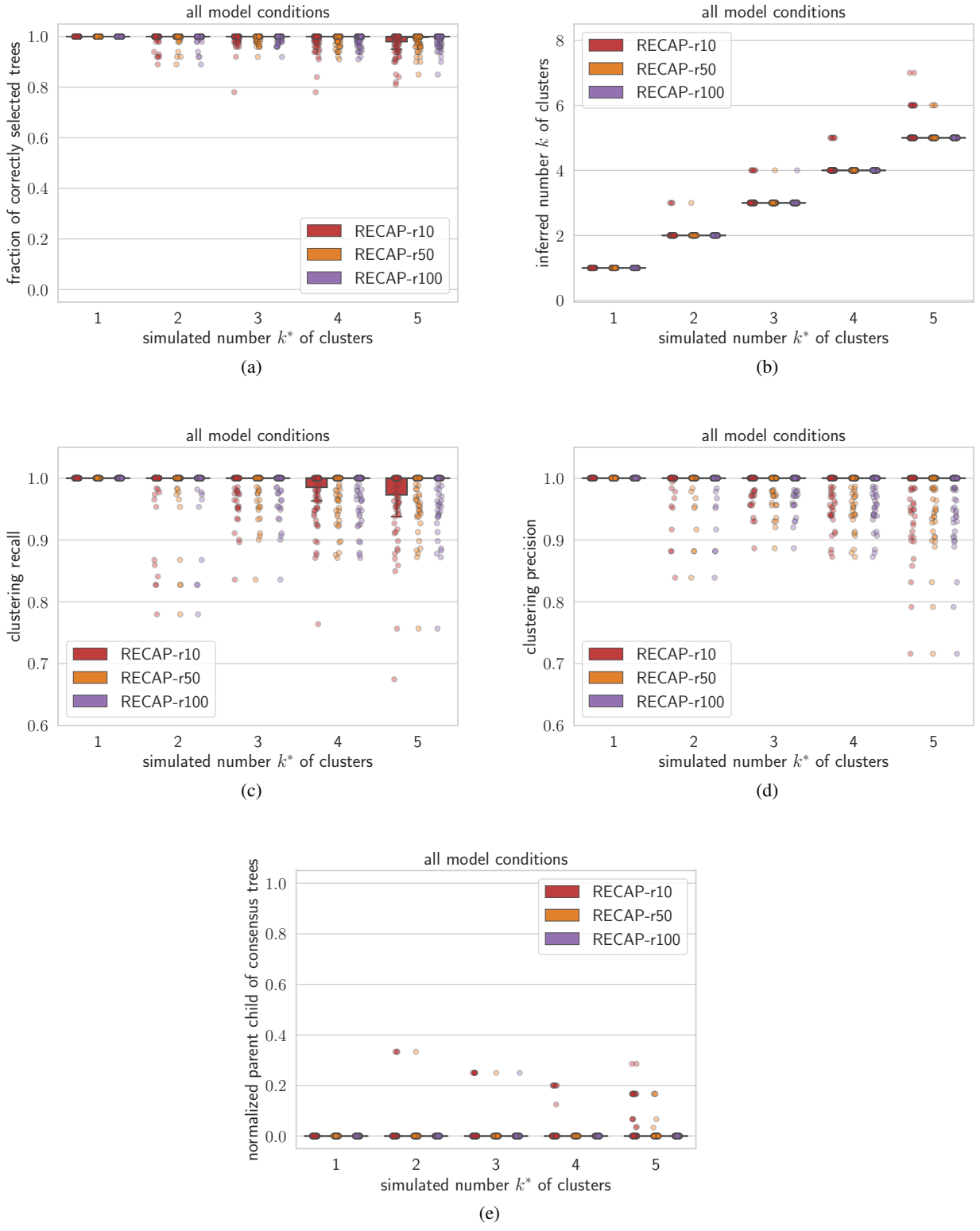
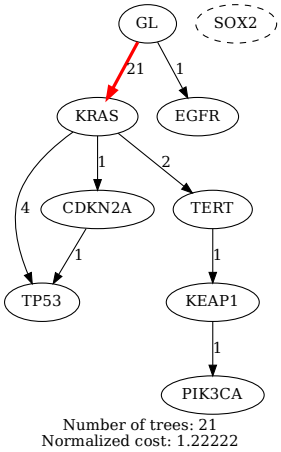
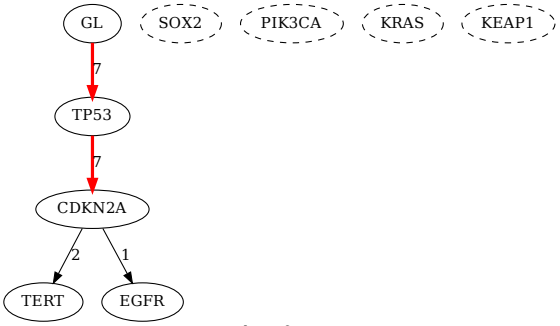


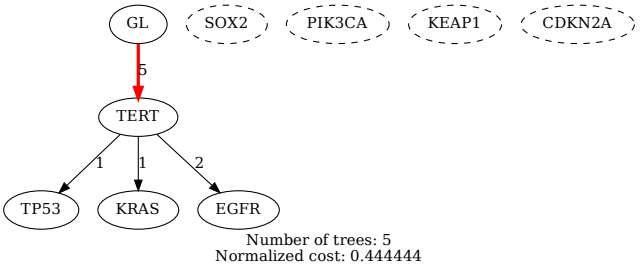
Fig S5: **Simulations show that performance of RECAP slightly increases with more restarts.** (a) The fraction of patients with correctly inferred trees. (b) The number k of patient clusters inferred. (c) The fraction of patient pairs that are correctly clustered together. (d) The fraction of patient pairs that are correctly put in separate clusters. (e) The normalized parent-child distances between the inferred and ground truth consensus trees.



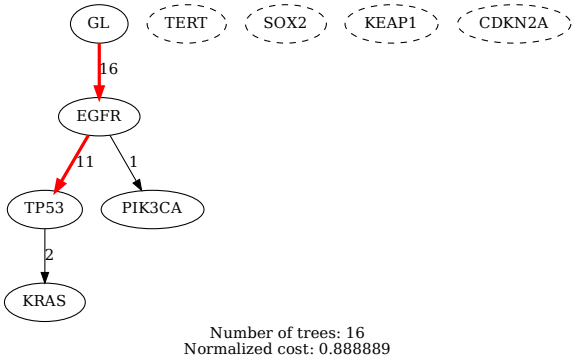
(a) Cluster 1



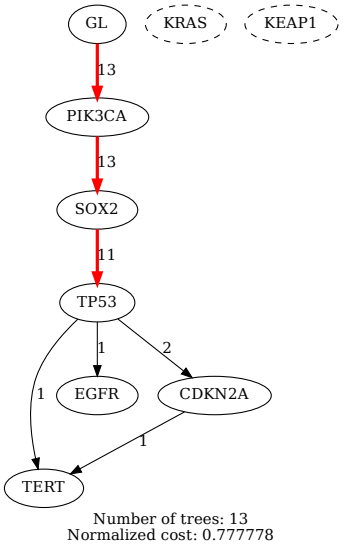
(b) Cluster 2



(c) Cluster 3

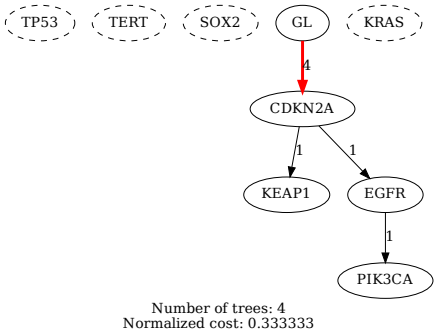


(d) Cluster 4

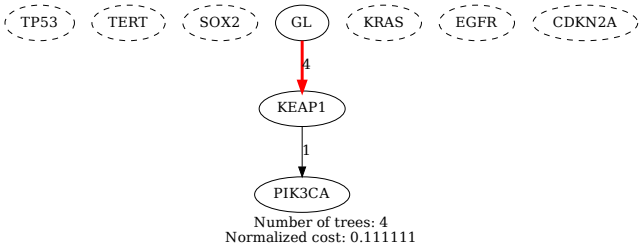


(e) Cluster 5

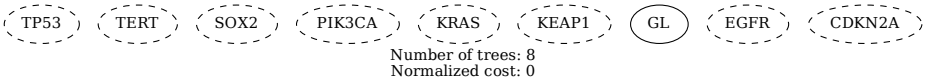
Fig S6: **Consensus trees identified by RECAP for a non-small cell lung cancer cohort (Jamal-Hanjani *et al.*, 2017).** Red edges indicate consensus tree edges, edge label indicates the number of patients that contain the edge. Dashed vertices indicate missing mutations. Continued in Fig. S7.



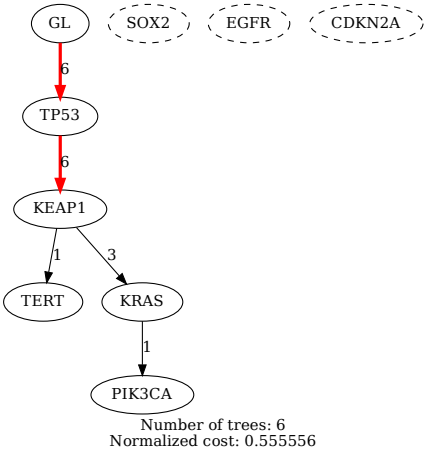
(a) Cluster 6



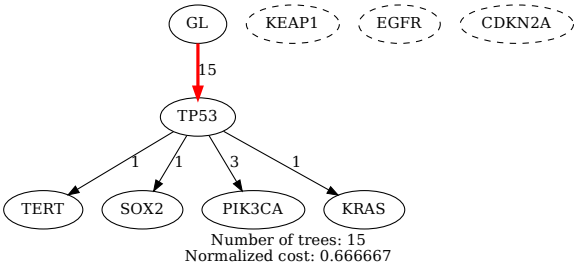
(b) Cluster 7



(c) Cluster 8

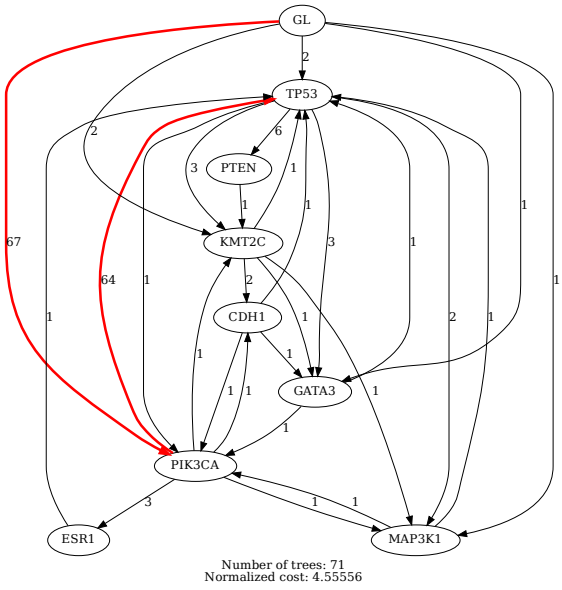


(d) Cluster 9

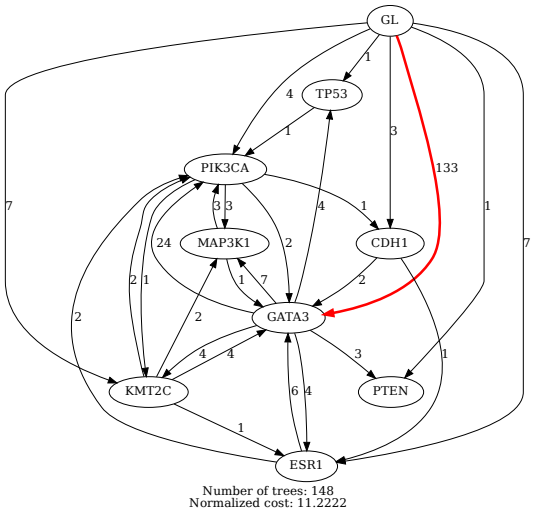


(e) Cluster 10

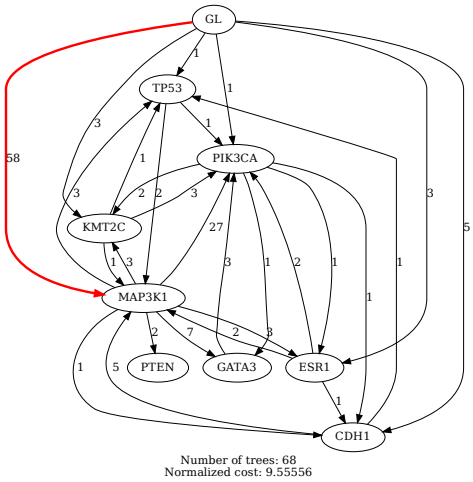
Fig S7: **Consensus trees identified by RECAP for a non-small cell lung cancer cohort (Jamal-Hanjani *et al.*, 2017).** Red edges indicate consensus tree edges, edge label indicates the number of patients that contain the edge. Dashed vertices indicate missing mutations. Note that Cluster 8 corresponds to the empty consensus tree, comprised of only the germline vertex. Continued from Fig. S6.



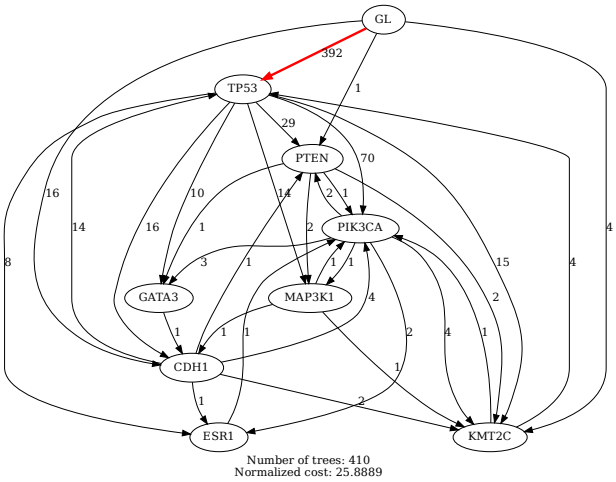
(a) Cluster 1



(b) Cluster 2

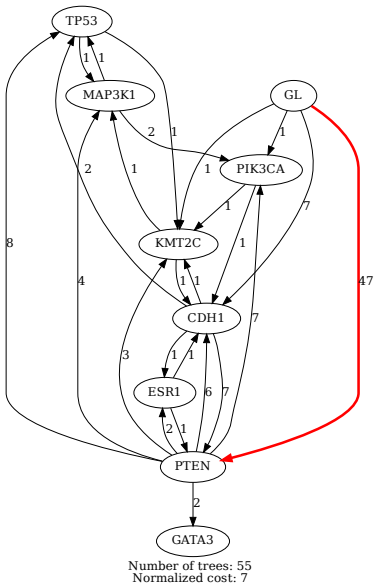


(c) Cluster 3

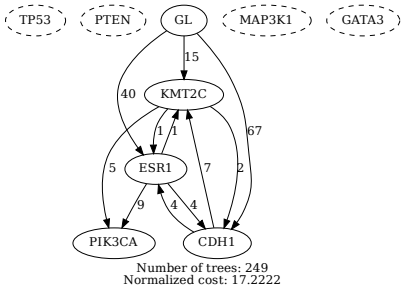


(d) Cluster 4

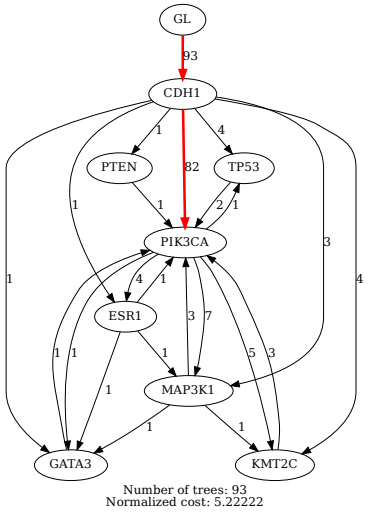
Fig S8: **Consensus trees identified by RECAP for a breast cancer cohort (Razavi *et al.*, 2018).** Red edges indicate consensus tree edges, edge label indicates the number of patients that contain the edge. Dashed vertices indicate missing mutations. Continued in Fig. S9.



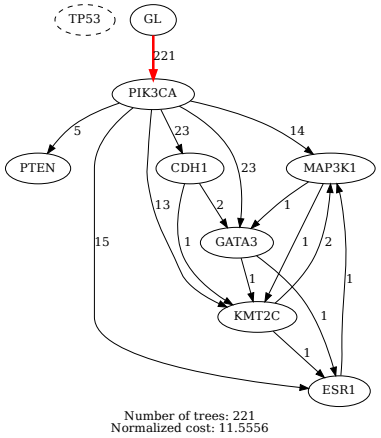
(a) Cluster 5



(b) Cluster 6



(c) Cluster 7



(d) Cluster 8

Fig S9: **Consensus trees identified by RECAP for a breast cancer cohort (Razavi *et al.*, 2018).** Red edges indicate consensus tree edges, edge label indicates the number of patients that contain the edge. Dashed vertices indicate missing mutations. Continued from Fig. S8.