Characterizing the Solution Space of Migration Histories of Metastatic Cancers with MACH2

Mrinmoy S. Roddur[1], Vikram Ramavarapu[1], Abigail Bunkum[3], Ariana Huebner[3], Roman Mineyev[1], Nicholas McGranahan[3], Simone Zaccaria[3,†], and Mohammed El-Kebir[1,2,†]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, IL 61801, USA
[2]Cancer Center at Illinois, University of Illinois at Urbana-Champaign, IL 61801, USA
[3]University College London Cancer Institute, London, United Kingdom
[†]Correspondence: s.zaccaria@ucl.ac.uk, melkebir@illinois.edu

Abstract

Understanding the migration history of cancer cells is essential for advancing metastasis research and developing therapies. Existing migration history inference methods often rely on parsimony criteria such as minimizing migrations, comigrations, and seeding locations. Importantly, existing methods either yield a single optimal solution or are probabilistic algorithms without guarantees on optimality nor comprehensiveness of the returned solution space. As such, current methods are unable to capture the full extent of uncertainty inherent to the data. To address these limitations, we introduce MACH2, a method that systematically enumerates all plausible migration histories by exactly solving the PARSIMONIOUS MIGRATION HISTORY WITH TREE REFINEMENT (PMH-TR). In addition to the migration and comigration criteria, MACH2 employs a novel parsimony criterion that minimizes the number of clones unobserved in their inferred location of origin. MACH2 allows one to specify both the order and the set of criteria to include during optimization, allowing users to adapt the model to specific analysis needs. MACH2 also includes a summary graph to identify high-confidence migrations. Finally, we introduce MACH2-viz, an interactive webtool for visualizing and exploring MACH2 solution spaces. Using simulated tumors with known ground truth, we show that MACH2, especially the version that prioritizes the new unobserved clone criterion, outperforms existing methods. On real data, MACH2 detects extensive uncertainty in non-small cell lung, ovarian, and prostate cancers, and infers migration histories consistent with orthogonal experimental data.

1 Introduction

Metastasis, the spread of cancer cells from the primary tumor to distant anatomical locations, is the main cause of cancer-related deaths [1,2]. Reconstructing the precise pattern of spread of metastatic cancers helps uncover metastatic pathways, identify treatment targets, and develop precision medicine [3–11]. Since cancer cells in different locations evolve independently, one way to trace the migration history of a tumor is by analyzing the evolutionary relationships of its cells across metastatic sites. To that end, researchers run computational methods [12–18] to identify *clones* — groups of cells with the same set of mutations — from sequencing data of primary and metastatic biopsies as well as their evolutionary history in the form of a *clonal tree*, where each node represents a clone labeled by the anatomical locations in which it was observed (Fig. 1a,b). Recent migration history inference methods further analyze this clonal tree to determine each clone's location of origin and identify migrations indicated by parent-child pairs with distinct origins via one or more parsimony criteria [19–21]. The key principle underlying these methods is that metastatic spread between locations is a rare event, and therefore inferring migration histories with as few metastasis events is preferable over more complex histories with many such events.

The *migration criterion*, which minimizes the number of migrations was first proposed by Slatkin and Madisson [22] and later, used by McPherson et al. [23] to reconstruct migration histories of metastatic ovarian cancers. Extending this work, the MACHINA algorithm [19] proposed to solve the PARSIMONIOUS MIGRATION HISTORY WITH TREE REFINEMENT (PMH-TR) problem (Fig. 1c), which included two additional criteria: (i) the *comigration criterion*, minimizing the number of comigrations, or simultaneous migrations of one or more clones from one location to another [19, 24–33], and (ii) the *seeding location criterion*, minimizing the number of locations that seed other metastases. The PMH-TR problem also addresses uncertainty reflected by *polytomies*, i.e. nodes with more than two children. While each cell division results in two daughter cells, the input clonal tree might be non-binary and contain polytomies typically due to data limitations such as limited sequencing depth. A subset of polytomies can be refined by introducing additional nodes and edges guided by minimization of the parsimony criteria (Fig. S1). In recent work, Roddur et al. [34] modified the comigration criterion to ensure that comigrations maintain temporal consistency with respect to the output clonal tree.

One key limitation of MACHINA is that it produces a single optimal solution to the PMH-TR problem, overlooking alternative explanations for metastatic spread that are equally supported by the data. Importantly, not taking into account the full solution space, and focusing on a single solution, underestimates the extent of uncertainty and may lead to incorrect conclusions about metastatic spread. A second limitation of MACHINA is that it adheres to a fixed criteria ordering: first minimizing migrations, then comigrations, and finally seeding locations. A recent method, Metient [20] overcomes these two issues by returning multiple solutions minimizing the three discussed criteria in any of six possible permutations. Metient ranks the identified solutions by learning criterion-specific weights across a patient cohort using prior information in the form of the branch lengths/genetic distances of migration edges and, optionally, organotropism information (prior probabilities of directional seeding between a pair of locations). The stochastic gradient descent approach employed by Metient offers no guarantees regarding optimality of the identified solutions nor guarantees about the completeness of the reconstructed solution space. This combined with the requirement for a homogeneous cohort of patients to properly calibrate the criterion-specific weights, potential unreliability in genetic distances, and the seeding location criterion not being realistic for many cancer types, means that Metient, similarly to MACHINA, may overlook alternative parsimonious solutions supported by the data.

Addressing the limitations of prior methods, we introduce MACH2 (Migration Analysis of Clonal Histories 2), a new method that guarantees to enumerate all optimal solutions to the PMH-TR problem. Analogous to the unsampled lineage criterion used in phylodynamics [35–38], MACH2 introduces the *unobserved clone criterion*, minimizing the number of clones assigned to locations in which they were not observed (Fig. S1). Rather than adhering to a specific fixed ordering, MACH2 supports the specification of arbitrary criteria orderings of unobserved clones, migrations, and comigrations. In addition, MACH2 ensures the inferred migrations to be temporally consistent [34]. To efficiently enumerate the complete PMH-TR optimal solution space, MACH2 employs a new compact representation of refinements as edge-weighted multigraphs (Fig. S2). MACH2 represents the enumerated solution space in the form a summary graph, allowing one to identify high-confidence migrations (Fig. 1d). Alongside MACH2, we introduce MACH2-viz, an interactive tool for visualizing MACH2's solution space. On the theoretical side, we provide new complexity results for the decision and counting versions of the PMH-TR problem. Running MACH2 on simulated data with different criteria orderings, along with MACHINA and Metient, we found minimizing the number unobserved clones first, followed by the number of migrations and comigrations yielded the best results, and made this ordering the default recommendation for MACH2. Moreover, we found MACH2 to identify extensive uncertainty in

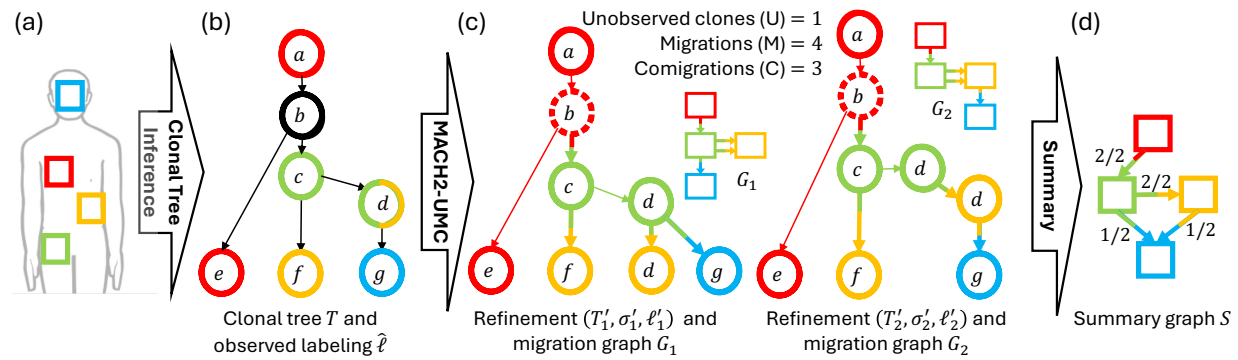


Figure 1: **Overview of MACH2.** (a) From sequencing data of matched primary and metastatic biopsies, (b) we infer a clonal tree T with observed location labeling $\hat{\ell}$ (colors), describing the evolutionary history of the tumor and the clonal composition of each biopsy, respectively. Here, clone 'b' was not observed in any location, whereas clone 'd' was observed in the green and yellow locations. (c) MACH2 in UMC mode enumerates all optimal PARSIMONIOUS MIGRATION HISTORY WITH TREE REFINEMENT (PMH-TR) solutions composed of refinements of $(T, \hat{\ell})$, lexicographically minimizing the number of unobserved clones (U, indicated by dashed circles), then the number of migrations (M, indicated by bichromatic edges), and finally, the number of comigrations (C). Here, there are two distinct optimal solutions with different patterns of migration, as shown by their migration graphs (rectangular nodes). (d) The corresponding summary graph of all migration graphs.

real data of metastatic non-small cell lung [39], ovarian [23] and prostate [40] cancers, with the input data supporting many alternative migration histories. In many cases, MACH2's solutions are more consistent with unused, orthogonal data compared to Metient and MACHINA.

2 Problem Statement

Our input is a clonal tree inferred by existing methods [12–18] from single-cell and/or bulk DNA sequencing data of matched primary and metastasis samples. Mathematically, a *clonal tree* is a tree T rooted at $r(T)$ with node set $V(T)$, leaf set $L(T)$ and edge set $E(T)$. The nodes $V(T)$ correspond to clones and each edge $(u, v) \in E(T)$ is directed such that u is closer to the root $r(T)$ than v. We denote the parent and the set of children of any node v by $\pi_T(v)$ and $\delta_T(v)$, respectively. We write $u \preceq_T v$ if node u is ancestral to v, i.e. there is a directed path from u to v. Conversely, we write $u \perp_T v$ if u and v occur on distinct lineages, i.e. neither $u \preceq_T v$ nor $v \preceq_T u$. We note that while clonal tree inference methods [12–18] assign mutations to clones, this information will not be used in the problem formulation and the resulting method MACH2. In addition to a clonal tree T, we are also given an *observed labeling* $\hat{\ell} : V(T) \to 2^{|\Sigma|}$ that maps each node $u \in V(T)$ to a subset $\hat{\ell}(u)$ of zero or more locations Σ in which the clone was observed (Fig. 1b). By definition, a leaf clone does not have any descendants, and for it to be inferred, it must be observed in at least one location. This leads to the following definition of an observed labeling.

Definition 1. *An* observed labeling *is a function $\hat{\ell} : V(T) \to 2^{|\Sigma|}$ that maps the clones $V(T)$ to subsets of locations Σ such that each leaf is observed in at least one location, i.e. $|\hat{\ell}(v)| \geq 1$ for all leaves $v \in L(T)$.*

There are three sources of uncertainty regarding a tumor's migration history. First, if $|\hat{\ell}(u)| = 0$ then node u was not observed in any location at the time of sequencing. Second, if $|\hat{\ell}(u)| > 1$ then node u was present in multiple locations and the precise migration order among these locations is unknown. Third, tree T may include *polytomies*, i.e. nodes with more than two children. To precisely specify the migration history, we need to resolve these sources of uncertainty (Fig. S1). To that end, we seek to refine $(T, \hat{\ell})$ into (T', σ', ℓ'). Tree T' is the refined clonal tree obtained from T as indicated by the refined-node to original-node mapping $\sigma' : V(T') \to V(T)$. The function $\ell' : V(T') \to \Sigma$ is a location labeling, assigning each node of T' exactly one location. We define refinements as follows.

Definition 2. *The tuple $(T', \sigma' : V(T') \to V(T), \ell' : V(T') \to \Sigma)$ forms a* refinement *of $(T, \hat{\ell} : V(T) \to 2^{|\Sigma|})$ provided (i) for each original node $v \in V(T)$, refined nodes $\{v' \in V(T') \mid \sigma'(v') = v\}$ induce a connected subtree in T', (ii) for each original edge $(u, v) \in E(T)$, there exists exactly one refined edge $(u', v') \in E(T')$ such that $\sigma'(u') = u$ and $\sigma'(v') = v$, (iii) for any two distinct, refined nodes $u', v' \in V(T')$ where $\sigma'(u') = \sigma'(v')$, it holds that*

$\ell'(u') \neq \ell'(v')$ and (iv) for each original node $v \in V(T)$ observed in location $s \in \hat{\ell}(v)$, there exists a refined node v' in T' such that $\sigma'(v') = v$ and $\ell'(v') = s$.

The mapping function σ' maps each refined node u' of T' to its corresponding original node $\sigma(u')$ in T, with conditions (i) and (ii) preserving original evolutionary relationships in the refined clonal tree T', condition (iii) preventing multiple, superfluous edges and migrations of the same original clone to the same location and condition (iv) enforcing the labeling ℓ' of T' to agree with the observed labeling $\hat{\ell}$ of T. In contrast with species phylogenetics, where refinements are typically defined for leaf-labeled trees (with leaves corresponding to taxa) our definition of refinement applies to node-labeled trees (where nodes correspond to clones) and includes additional constraints based on the input and output location labelings. In particular, the refined tree T' need not have the same set of leaves as the original tree T. We refer to Fig. 1c for two example refinements. While there exist many refinements for an input clonal tree T and observed labeling $\hat{\ell}$, we prioritize output refinements (T', σ', ℓ') using three different criteria.

First, we count the number $U(T', \sigma', \ell')$ of unobserved clones, which are refined nodes that are assigned locations they were not observed in. When a clone is assigned a location where it is not observed, one makes an inference not directly supported by the data. Thus, minimizing this criterion improves data fit (Fig. S1). We note that this criterion is analogous to the unsampled lineage criterion in phylodynamics [35–38].

Definition 3. *A refined node $v' \in V(T')$ is an* unobserved clone *if $\ell'(v') \notin \hat{\ell}(\sigma(v'))$, and $U(T', \sigma', \ell') = |\{v' \in V(T') \mid \ell'(v') \notin \hat{\ell}(\sigma(v'))\}|$ is the number of unobserved clones.*

Second, we count the number $M(T', \sigma', \ell')$ of migrations, defined as follows.

Definition 4. *A refined edge $(u', v') \in E(T')$ is a* migration *if $\ell'(u') \neq \ell'(v')$, and $M(T', \sigma', \ell') = |\{(u', v') \in E(T') \mid \ell'(u') \neq \ell'(v')\}|$ is the number of migrations.*

Third, we count the number $C(T', \sigma', \ell')$ of comigrations, which are defined as follows.

Definition 5. *A subset D of migrations is a* comigration *provided $\ell'(u') = \ell'(u'')$ and $\ell'(v') = \ell'(v'')$ for any two migrations $(u', v'), (u'', v'') \in D$.*

While our clonal trees T' lack exact timestamps, its edges $E(T')$ allow one to temporally order comigrations.

Definition 6. *Comigration D_p* precedes *comigration D_q, denoted as $D_p \prec_{\text{comig}} D_q$, if there exist migrations $(u_p, v_p) \in D_p$ and $(u_q, v_q) \in D_q$ inducing a directed path from u_p to v_q in T', i.e. $v_p \preceq_{T'} u_q$.*

Following [34], we ensure temporal consistency and define the number $C(T', \sigma', \ell')$ of comigrations as follows.

Definition 7. *A partition $\mathcal{D}$ of all migrations of a refinement (T', σ', ℓ') forms* consistent comigrations *provided (i) each part $D \in \mathcal{D}$ is a comigration and (ii) $(\mathcal{D}, \prec_{\text{comig}})$ is a partially ordered set.*

Definition 8. *The number $C(T', \sigma', \ell')$ is the size of the smallest partition of consistent comigrations.*

While the exact computation of $C(T', \sigma', \ell')$ is NP-hard, it can be efficiently determined for practical problem instances of moderate sizes using integer linear programming (ILP) [34]. The difference between the number of migrations and comigrations is that the former counts the total number of times an individual clone migrates whereas the latter counts the number of migration events composed of one or more clones. We refer to Fig. 1 and Fig. S1 for examples of scoring refinements using the three criteria discussed above. As discussed previously for migrations and comigrations [41] and illustrated in Fig. S1, there exist tradeoffs between the criteria, i.e. optimizing one criterion often comes at the expense of other criteria. We propose to model the desired tradeoff by taking as input an ordered subsequence $\mathcal{F} = (f_1, \ldots, f_k)$ of the aforementioned three criteria. We say that a solution (T', σ', ℓ') *lexicographically dominates* solution (T'', σ'', ℓ'') if there exists a criterion $j \in [k]$ for which (T'', σ'', ℓ'') received a higher cost than (T', σ', ℓ'), i.e. $f_j(T'', \sigma'', \ell'') > f_j(T', \sigma', \ell')$, while having the same costs in all previous criteria $1 \leq i < j \leq k$, i.e. $f_i(T', \sigma', \ell') = f_i(T'', \sigma'', \ell'')$ for all $i \in \{1, \ldots, j-1\}$. Then, the set of *lexicographically optimal refinements* consists of refinements for which no lexicographically dominating refinement exists. In this work, we focus on the following problem.

Problem 1 (PARSIMONIOUS MIGRATION HISTORY WITH TREE REFINEMENT (PMH-TR))**.** *Given a clonal tree T, set Σ of locations, observed labeling $\hat{\ell} : V(T) \to 2^{|\Sigma|}$ and criteria ordering $\mathcal{F}$, enumerate the space* PMH-TR$(\mathcal{F}, T, \hat{\ell})$ *of all refinements (T', σ', ℓ') of $(T, \hat{\ell})$ that are lexicographically optimal with respect to $\mathcal{F}$.*

In this work, we allow the criteria ordering $\mathcal{F}$ to correspond to any of the $\binom{3}{1}1! + \binom{3}{2}2! + \binom{3}{3}3! = 15$ ordered subsequences of the three criteria defined above (unobserved clones denoted as 'U', migrations as 'M', and comigrations as 'C'). For example, we denote with $\mathcal{F} = \mathrm{UMC}$ the criteria ordering where first unobserved clones are prioritized, followed by migrations and finally comigrations. While an expert user can specify any of the 15 possible criteria orderings, we set $\mathcal{F} = \mathrm{UMC}$ as the default recommended criteria ordering based on an extensive simulation study (Section 5.1). The original MACHINA [19] algorithm returns a single optimal solution for the $\mathcal{F} \in \{\mathrm{MC}, \mathrm{CM}\}$ cases.

3 Combinatorial Characterization

Here, we discuss properties of the optimal solution space of PMH-TR. Due to space constraints proofs have been moved to Appendix B.3. Existing methods, including MACHINA [19] and Metient [20], aim to refine leaf-labeled trees rather than fully-labeled trees. As such, these methods employ a preprocessing step to "pull down" leaves, i.e. attach new leaves to some of the existing nodes of the input clonal tree T, assigning each leaf a unique location. This procedure results in a new clonal tree $\overline{T}$ with leaf labeling $\overline{\ell} : L(\overline{T}) \to \Sigma$ such that there exists a bijection with the pairs (v, s) of nodes v observed in location $s \in \hat{\ell}(v)$ in the original clonal tree T. To that end, for each observed node-location pair (v, s) (such that v is an internal node or a leaf observed in more than one location, i.e. either $v \in I(T)$ or $v \in L(T)$ and $|\ell(v)| > 1$), we add a new leaf v' and edge (v, v') labeled by $\overline{\ell}(v') = s$. In addition, for each leaf v of T observed in exactly one location (i.e. $\hat{\ell}(v) = \{s\}$), we label v with $\overline{\ell}(v) = s$ in $\overline{T}$. We denote this procedure as $\textsc{PullDownLeaves}(T, \hat{\ell})$, yielding the pair $(\overline{T}, \overline{\ell})$. Although only the leaves have observed locations (in fact, in exactly one location), the resulting pair $(\overline{T}, \overline{\ell})$ itself is an instance of PMH-TR. We have the following important proposition, showing the redundancy of this preprocessing step.

Proposition 1. *Let* $(\overline{T}, \overline{\ell}) = \textsc{PullDownLeaves}(T, \hat{\ell})$ *and* $\mathcal{F} \in \{\mathrm{MC}, \mathrm{CM}\}$. *Then, the solution space* $\text{PMH-TR}(\mathcal{F}, T, \hat{\ell})$ *is a subset of the solution space* $\textsc{FromPulledDownRefinement}(\text{PMH-TR}(\mathcal{F}, \overline{T}, \overline{\ell}))$.

Corollary 2. *Let* $(\overline{T}, \overline{\ell}) = \textsc{PullDownLeaves}(T, \hat{\ell})$ *and* $\mathcal{F} \in \{\mathrm{MC}, \mathrm{CM}\}$. *Then, all the optimal solutions* (T', σ', ℓ') *and* $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ *to* PMH-TR *instances* $(\mathcal{F}, T, \hat{\ell})$ *and* $(\mathcal{F}, \overline{T}, \overline{\ell})$*, respectively, have the same number of migrations and comigrations, i.e.* $M(T', \sigma', \ell') = M(\overline{T}', \overline{\sigma}', \overline{\ell}')$ *and* $C(T', \sigma', \ell') = C(\overline{T}', \overline{\sigma}', \overline{\ell}')$.

We prove this by defining an injective function $\textsc{ToPulledDownRefinement}$ (Algorithm S2) to map each optimal refinement (T', σ', ℓ') of $(T, \hat{\ell})$ to a refinement $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ of $(\overline{T}, \overline{\ell})$, and a surjective function $\textsc{FromPulledDownRefinement}$ (Algorithm S3) to map each optimal refinement $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ of $(\overline{T}, \overline{\ell})$ to a refinement (T', σ', ℓ') of $(T, \hat{\ell})$. These two functions are defined more precisely in Appendix 3 and are illustrated in Fig. S2. Fig. S11 gives an example of solutions in $\text{PMH-TR}(\mathcal{F}, \overline{T}, \overline{\ell})$ that are not present in $\text{PMH-TR}(\mathcal{F}, T, \hat{\ell})$. To support efficient exhaustive enumeration of the PMH-TR solution space, we present an equivalent representation of refinements as constrained, directed multigraphs defined as follows.

Definition 9. *A directed multigraph* R *whose nodes* $V(R)$ *correspond to locations* Σ *and whose edges are labeled by* $e : E(R) \to V(T) \cup E(T)$ *is a* refinement graph *for a clonal tree* T *and observed labeling* $\hat{\ell} : V(T) \to 2^{|\Sigma|}$ *provided*

(i) for each node $u \in V(T)$*, the subgraph* R_u *of* R *is a directed tree without any multi-edges — where* R_u *is composed of: (i-a) all nodes* $s \in V(R)$ *with an incoming edge* $(s', s) \in E(R)$ *labeled by either* $e((s', s)) = u$ *or* $e((s', s)) = (\pi_T(u), u)$ *or an outgoing edge* $(s, t) \in E(R)$ *labeled by either* $e((s, t)) = u$ *or* $e((s, t)) = (u, v)$ *for any* $v \in \delta_T(u)$*, and (i-b) all edges* $(s, t) \in E(R)$ *labeled by* $e((s, t)) = u$*;*

(ii) for each edge $(u, v) \in E(T)$*, there is exactly one edge* $(s, t) \in E(R)$ *labeled by* $e((s, t)) = (u, v)$*;*

(iii) for each node $u \in V(T)$*,* $\hat{\ell}(u) \subseteq V(R_u)$*;*

(iv) for each node $u \in V(T) \backslash \{r(T)\}$*,* R_u *is rooted at t if there exists an edge* (s, t) *labeled by* $e((s, t)) = (\pi_T(u), u)$.

Importantly, there exists a bijection between refinement graphs (R, e) and refinements (T', σ', ℓ') (Fig. S2).

Theorem 3. *There exists a bijection between refinements* (T', σ', ℓ') *and pairs* (R, e) *of edge-labeled refinement graphs for a given tree* T *with observed labeling* $\hat{\ell}$.

We prove Theorem 3 by showing that the function CONSTRUCTTREE$(T, \hat{\ell}, R, e)$ (Algorithm S4) returns a refinement (T', σ', ℓ') if and only if CONSTRUCTGRAPH$(T, \hat{\ell}, T', \sigma', \ell')$ (Algorithm S5) returns the refinement graph (R, e). The following corollary directly follows from procedures CONSTRUCTTREE and CONSTRUCTGRAPH and implies that for each edge (u, v) in T' there exists exactly one edge $(\ell'(u), \ell'(v))$ in (R, e).

Corollary 4. *For any refinement (T', σ', ℓ') of instance $(T, \hat{\ell})$, there is a bijection between the edges in T' and the edges in the corresponding refinement graph R.*

We list the properties of the optimal solution space in relation to different criteria ordering in Appendix B.2. In addition, we prove the decision and counting versions of PMH-TR to be NP-complete and #P-complete for several criteria orderings, including the default ordering $\mathcal{F} = \mathrm{UMC}$ (Appendix B.4 and summarized in Fig. S2c).

4 Method

In this section, we introduce MACH2 (Migration Analysis of Clonal Histories 2), an exact algorithm to solve the PMH-TR problem (Section 4.1) given a clonal tree with observed labeling. To interpret the set of refinements inferred by MACH2, we define a migration graph and summary graph, and introduce MACH2-viz in Section 4.2.

4.1 Integer Linear Program for PMH-TR

Since PMH-TR is both NP-hard and #P-hard (Appendix B.4), and therefore is unlikely solvable in polynomial time, MACH2 employs an integer linear program (ILP) based on the compact, equivalent representation of solutions with refinement graphs R and edge labelings e (Section 3). Our ILP is carefully constructed to maintain a bijection between its optimal solutions and the PMH-TR solution space. The ILP has $\mathcal{O}(|\Sigma|^2|V(T)|)$ variables (with the same asymptotic number of integer variables) and $\mathcal{O}(|\Sigma|^3|V(T)|)$ constraints. We provide a complete description of the ILP in Appendix C. MACH2 provides an implementation of the ILP using Gurobi, and is publicly available at `https://github.com/elkebir-group/MACH2` under the BSD 3-clause license.

4.2 Interpretation of MACH2 Solution Spaces

To interpret the refinements (T', σ', ℓ') and analyze the corresponding migration histories in Section 5, we introduce several concepts and a visualization tool. First, following El-Kebir et al. [19], we depict the migration history associated with a refinement (T', σ', ℓ') by a *migration graph*, defined as follows.

Definition 10 (El-Kebir et al. [19])**.** *A directed multigraph G is a* migration graph *for a PMH-TR solution (T', σ', ℓ') provided (i) the node set $V(G)$ corresponds to the input locations Σ and (ii) for each edge (s, t) in the multi-set $E(G)$ of edges there is a unique migration (u', v') such that $\ell'(u') = s$ and $\ell'(v') = t$.*

Note that a migration graph G' for a refinement (T', σ', ℓ') is obtained by removing self-loops from corresponding refinement graph R introduced in Section 3. We show examples of migration graphs in Fig. 1c. To summarize the solution space of PMH-TR instance $(\mathcal{F}, T, \hat{\ell})$, we introduce the summary graph, defined as follows.

Definition 11. *An edge-weighted, directed graph S is a* summary graph *of a solution space PMH-TR$(\mathcal{F}, T, \hat{\ell})$ provided (i) the node set $V(S)$ corresponds to the input locations Σ and (ii) S has an edge (s, t) with weight $w(s, t) > 0$ if and only if there exist $w(s, t) > 0$ solutions $(T', \sigma', \ell') \in$ PMH-TR$(\mathcal{F}, T, \hat{\ell})$ with a migration (u', v') from location $\ell'(u') = s$ to location $\ell'(v') = t$.*

Often we divide the weights by the total number of solutions to get the probability of a migration happening between a pair of locations for PMH-TR$(\mathcal{F}, T, \hat{\ell})$. Summary graphs help in quantifying the likelihood of migrations between specific locations and also facilitate focusing on a subset of solutions with specific migrations. We show an example of a summary graph in Fig. 1d. As discussed, both MACHINA and Metient use the seeding location criterion, which minimizes the number of seeding locations defined as follows.

Definition 12 (El-Kebir et al. [19])**.** *A location $s \in \Sigma$ is a* seeding location *if there is a migration (u', v') i.e. $\ell'(u') \neq \ell'(v')$ such that $\ell'(u') = s$.*

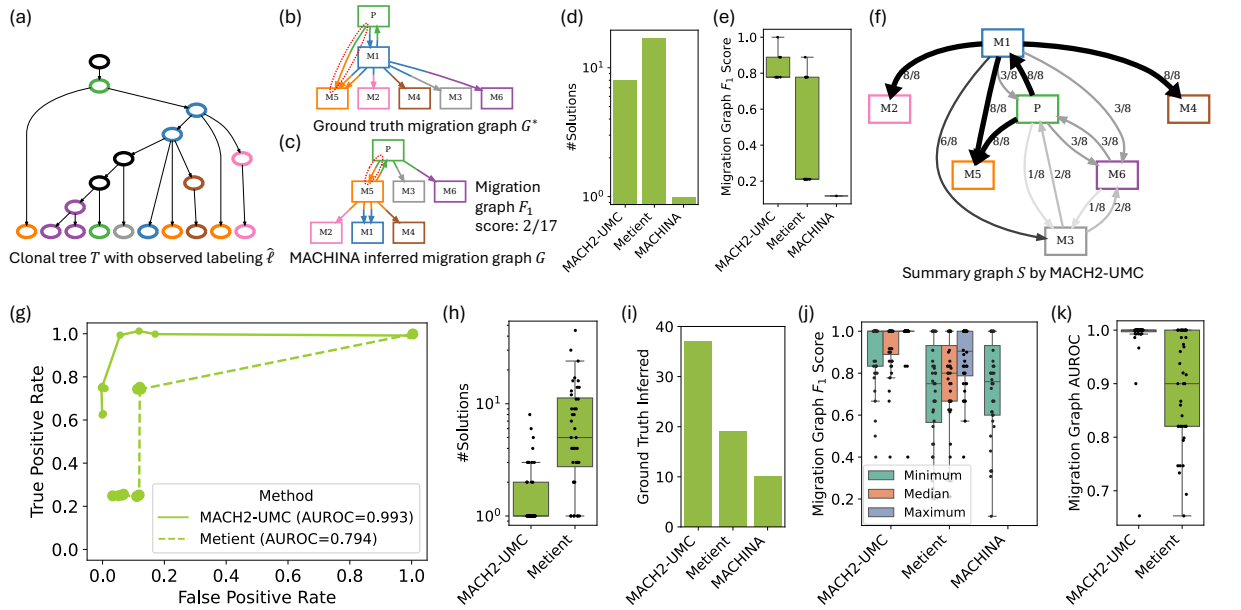


Figure 2: **Comparison between MACH2-UMC, Metient [20] and MACHINA [19] on simulated data.** (a) Example simulation instance $(T, \hat{\ell})$ and (b) ground-truth migration graph G^*. (c) The migration graph reported by MACHINA, achieving a migration graph F_1 score of $2/17 \approx 0.118$ (with dashed ellipses showing matching edges). (d) The number of identified solutions of the illustrated simulation instance. (e) The summary graph of the 8 solutions identified by MACH2-UMC (with bold edges indicating migrations present in all solutions). (f) Migration graph F_1 score of the illustrated simulation instance. (g) ROC curves obtained from summary graphs of the shown simulation instance. (h) MACH2 and Metient inferred multiple solutions for majority of the 40 simulation instances. (i) Number of times each method included the ground-truth solution. (j) Minimum, median, and maximum migration graph F_1 score across all instances (with a single instance shown in Fig. S4b). (k) Distribution of AUROC values across all instances (with a single instance shown in (f)).

Although MACH2 does not directly incorporate the seeding location criterion into its problem formulation as it is not realistic for several cancer types, it retains the criterion as an optional filtering step to prioritize solutions if desired. Finally, we present MACH2-viz, a visual tool for exploring PMH-TR solution spaces (Fig. S3). The tool is accessible from MACH2 itself, or through the deployed web application via `https://elkebir-group.github.io/mach2-viz/`. MACH2-viz is written in JavaScript using React and Cytoscape libraries. The interface allowing users to open the visualizer directly from MACH2 is written in Python, using the Flask library. We refer to Appendix D for more details.

5 Results

5.1 Simulations

To benchmark MACH2, we compared it to Metient [20] and MACHINA [19] on 40 previously-published simulation instances [19] with $|\Sigma| \in \{5, \ldots, 7\}$ locations (referred to as 'm5' in [19]). These instances were generated using an agent-based model accounting for tumor evolution, selection and migration. Importantly, the simulation instances contained clones that were not observed in any location (ranging from 5.88% to 26.32% of total clones), due to complete clonal expansions as well as limited-resolution sequencing. Each simulation instance consists of a clonal tree T with location labeling $\hat{\ell}$, which were provided as input to each method. In addition, we provided Metient with branch lengths corresponding to the number of mutations introduced along each edge. As in the original publication [20], we ran Metient in 'calibrate' mode on two cohorts, one cohort containing only primary-to-metastasis spread, and another also containing metastasis-to-metastasis spread. In this simulation study, we assess the number of solutions, whether the enumerated solution space included the ground-truth migration history, and the accuracy of the inferred migration histories and inferred migrating clones. For brevity, we will only report solutions obtained with MACH2 with criteria ordering $\mathcal{F} = \mathrm{UMC}$ (denoted MACH2-UMC), which outperformed all other criteria orderings (as discussed later).

As an example, we show a simulation instance $(T, \hat{\ell})$ with $|V(T)| = 19$ clones and $|\Sigma| = 7$ locations in Fig. 2a, and its corresponding ground-truth migration graph G^* in Fig. 2b. By design, MACHINA inferred only a single solution, which did not match the ground-truth migration graph (Fig. 2c). MACH2-UMC and Metient inferred 8 and 17 solutions, respectively, with the MACH2-UMC summary graph shown in Fig. 2d. While the MACH2-UMC solutions included the ground truth, the Metient solutions did not. To more precisely quantify how well the inferred migration graphs match the ground truth, we use the previously introduced *migration graph F_1 score* [19], which equals the harmonic mean of precision and recall of migration graph edges (see Appendix E). The migration graph G reported by MACHINA missed 8 out of 9 migration edges of G^* and included 7 migration edges not present in G^*, leading to a migration graph F_1 score of $2/17 \approx 0.118$. On the other hand, the 17 Metient solutions achieved migration graph F_1 scores ranging from 0.211 to 0.889 and a median of 0.211. Finally, the 8 MACH2-UMC solutions achieved migration graph F_1 scores ranging from 0.778 to 1 and a median of 0.778. Fig. 2e shows the distributions of the migration graph F_1 scores for this one simulation instance.

To deal with the uncertainty of multiple optimal solutions in real data, one could restrict downstream analyses to only those migrations between locations that have strong support among the enumerated solutions. In other words, for a specified threshold $\tau \in [0, 1]$, one could retain only those summary graph edges that are present in at least a fraction τ of the solution space. For the illustrated simulation instance, setting $\tau = 1$ restricts analysis to the 5 migrations present in all 8 solutions identified by MACH2-UMC (Fig. 2f). By varying τ we can assess the trade-off between sensitivity (or true positive rate) and specificity (or false positive rate) of each method in a Receiver Operating Characteristic (ROC) plot (Fig. 2g). For this instance, we find that MACH2-UMC achieved the largest Area Under the ROC (AUROC) values of 0.993, outperforming Metient (AUROC: 0.794).

When considering all 40 simulation instances, all methods, except MACHINA, inferred multiple solutions for the majority of simulation instances (MACH2-UMC ranging from 1 to 8 solutions and Metient from 1 to 45 solutions; Fig. 2h). MACH2-UMC inferred the ground truth 37 out of 40 times (Fig. 2i). On the other hand, Metient included the ground-truth solution only 19 times and MACHINA only 10 times. Furthermore, MACH2-UMC outperformed all other methods for the migration graph F_1 score, achieving median values of 1 for the minimum, median and maximum migration graph F_1 score (Fig. 2j). Metient performed worse than MACH2-UMC, achieving median values of 0.75 for the minimum, 0.8 for the median and 0.89 for the maximum of the computed migration graph F_1 scores across the identified solution spaces. MACHINA achieved median values of 0.75 for the migration graph F_1 score. Finally, when considering the AUROC values obtained by thresholding the summary graphs of the inferred solution spaces, we find MACH2-UMC performs best, achieving the largest total AUROC value of 39.46 (summed across all 40 simulation instances) whereas Metient achieved a total AUROC value of 35.57. We show the AUROC distributions in Fig. 2k.

To assess the impact of the choice of criteria ordering $\mathcal{F}$, we ran MACH2 with 13 different criteria orderings. We excluded MACH2-U and MACH2-C, as they were too unconstrained, leading to too many and often nonsensical solutions and excessive running times. We found that for all criteria ordering, MACH2 inferred multiple solutions for the majority of simulation instances (Fig. S4a). MACH2-UM included the ground-truth solution in its solution space 39 out of 40 times, followed by MACH2-UMC, which returned a subset of MACH2-UM solutions, inferring the ground truth 37 out of 40 times (Fig. S4b). Although both MACH2-UM and MACH2-UMC achieved median values of 1 for the minimum, median and maximum migration graph F_1 score, MACH2-UMC had the highest mean across all the criteria orderings for the minimum (0.902) and median (0.929) migration graph F_1 score (Fig. S4c). For AUROC values, both MACH2-UMC and MACH2-UM achieved a median value of 1 (Fig. S4d).

In summary, we found MACH2-UMC performed best with respect to other criteria orderings as well as MACHINA and Metient, demonstrating the importance of the new unobserved clones (U) criterion. In addition, we showed that ignoring alternative optimal solutions might lead to wrong migration history inferences. Finally, we demonstrated the utility of summary graphs to summarize large solution spaces and identify high-confidence migrations.

5.2 Application of MACH2 to Real Data

5.2.1 MACH2 Accurately Characterizes Uncertainty in Metastatic Spread

We applied MACH2 in UMC mode, Metient [20], and MACHINA [19] to three real datasets, including (i) one non-small-cell lung cancer patient (CRUK0063) from the TRACERx study [39], (ii) ten androgen-deprived metastatic prostate cancer patients [40], and (iii) seven high-grade serous ovarian cancer patients [23]. Each method was provided the same input clonal tree and observed labeling. In addition, Metient was provided the genetic distance of each edge as input and access to other instances of the same cohort to calibrate its parameters.

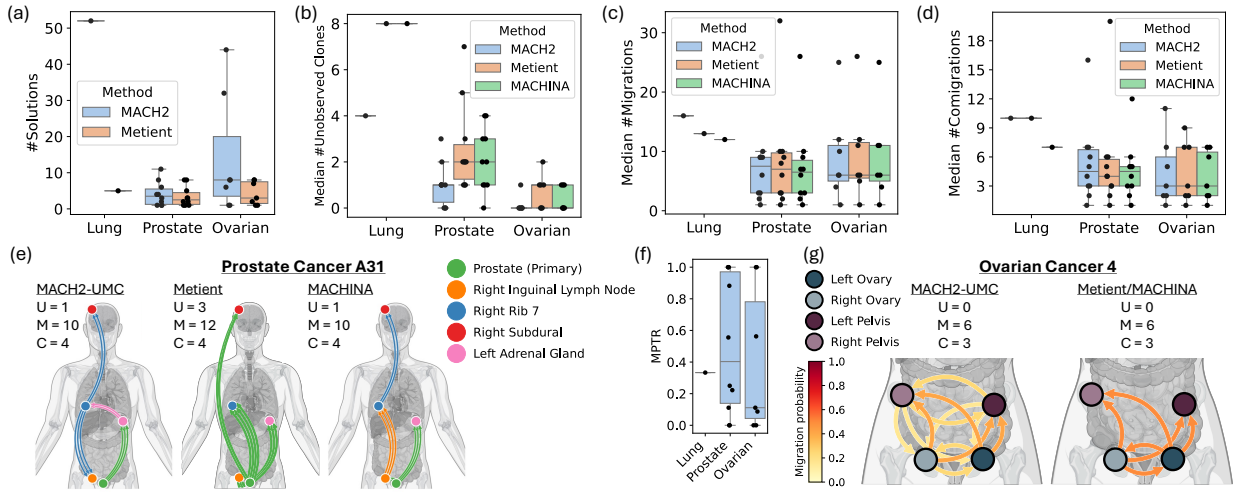


Figure 3: **Non-uniqueness of optimal migration histories identified by MACH2.** (a) MACH2 and Metient identify multiple solutions for metastatic non-small-cell lung [39], prostate [40] and ovarian [23] cancers. (b) MACH2 in UMC mode minimizes the number unobserved clones, while keeping the number of (c) migrations and (d) comigrations comparable to Metient and MACHINA. (e) For Prostate Cancer A31 [40], both MACH2 and Metient inferred solutions with migration from the primary location (prostate) to the right inguinal lymph node, which is consistent with prior studies [42, 43] unlike the solution returned by MACHINA, where 'right rib 7' seeded the right inguinal lymph node. (f) Migration pattern totality ratio. (g) The solution space of Ovarian Cancer 4 inferred by MACH2 had an MPTR of 0, indicating the complete lack of certainty in the pattern of metastatic spread (see Fig. S10). Due to the seeding location objective, Metient solutions inaccurately indicate only migrations from a primary ovary location to a metastatic site are plausible. U: unobserved clones, M: migrations, and C: comigrations.

For the majority of the real data instances, MACH2-UMC and Metient inferred multiple solutions, with 52 solutions for MACH2-UMC and 5 for Metient in the lung cancer instance, between 1 and 11 solutions for MACH2-UMC and between 1 and 8 for Metient across ten prostate cancer instances, and, between 1 and 44 solutions for MACH2-UMC and between 1 and 8 for Metient in seven ovarian cancer instances (Fig. 3a and Table S1). As shown in Fig. S6a, we found the number of MACH2-UMC inferred solutions to be slightly correlated with an increasing number of input clones (Pearson coefficient $r = 0.33$; t-test p-value $p = 0.01$) and locations ($r = 0.28$; $p = 0.04$). Since MACH2-UMC prioritizes minimizing the number of unobserved clones, it inferred solutions with the fewest unobserved clones across all 18 real instances compared to Metient and MACHINA (Fig. 3b and Table S1). On average, the median number of unobserved clones for MACH2 was 49.75% lower than Metient's and 37.5% lower than MACHINA's. Nevertheless, the median number of migrations and comigrations were similar across methods, with the number of migrations in MACH2 being 1.45% lower than Metient's and 5% higher than MACHINA's, while the number of comigrations were 5.17% and 10.93% higher in MACH2 than in Metient and in MACHINA, respectively (Fig. 3c-d and Table S1). In terms of running time, MACH2-UMC outperformed both Metient and MACHINA, with a median running time of 0.44 seconds, compared to 12.88 seconds for Metient and 2.09 seconds for MACHINA (Fig. S6b). MACH2 achieves this speed-up due to the compact graph-based representation of refinements as discussed in Section 3. We report the running time and peak memory usage in relation with the number of nodes, locations, and solutions in Fig. S7, showing increases in both quantities with increasing number of nodes, locations, and solutions.

Importantly, methods that generate multiple solutions are better at capturing alternative migration patterns, which may be overlooked by methods returning a single solution like MACHINA. Fig. 3e shows such an example, where for Prostate Cancer A31 [40] MACHINA yielded a single solution where the 'right rib 7' seeded the right inguinal lymph node, which is in close proximity to the prostate. In contrast, both MACH2 and Metient yielded solutions with the prostate seeding the right inguinal lymph node. This finding better aligns with prior studies suggesting that proximal lymph nodes near the prostate gland are often the initial location of metastasis, from which migrating clones further spread to distant sites [42, 43]. While MACH2 is guaranteed to infer temporally consistent comigrations [34], MACHINA and Metient may infer temporally inconsistent solutions (cf. Prostate Cancer A22 for Metient in Fig. S8).

Beyond including more biologically-justifiable migration histories, MACH2's ability to generate the complete solution space allows it to fully characterize the uncertainty inherent to migration history inference. This is a key ability that will allow one to avoid drawing incorrect conclusions due to not fully taking the inherent, data-specific

uncertainty in migration history inference into account. In such cases, summary graphs can be used to identify possible migrations along with probabilities. In Fig. S9a, the summary graph for Prostate Cancer A24 contained 2 migrations that were shared across all 8 solutions (prostate to right axillary lymph node, right diaphragm to right rib). Therefore, it can be said with greater certainty that these migrations likely occurred. Conversely, edges in the summary graph with probabilities below 1 indicate uncertainty about the corresponding migration. For example, the probability of xiphoid to be seeded from right axillary lymph node, right rib, or right diaphragm was $4/8 = 0.5$, $2/8 = 0.25$, and $2/8 = 0.25$, respectively. We propose to quantify this uncertainty using the *migration pattern totality ratio* (MPTR), defined as the ratio of the intersection to the union of migration graph edges among all solutions. As such, an MPTR of 0 indicates no shared migrations common to all solutions whereas an MPTR of 1 indicates that each migration is present in all solutions (details in Section E.3).

In our real data experiments, we found MPTR to decrease with increasing number of solutions (Fig. 3b and Table S1). Strikingly, we found instances where MPTR was 0, indicating the absence of shared migrations across all the solutions (Fig. 3f and Table S1). One such extreme case is Ovarian Cancer 4 [23] where MACH2-UMC predicted 32 different solutions, corresponding to all possible tree-like seeding patterns among the left ovary, right ovary, left pelvic area, and right pelvic area (Fig. S10). The corresponding summary graph in Fig. 3g predicts the probability of migration between any two pairs of locations to be nonzero, but less than one. Note that although Metient, like MACH2, returns multiple solutions, it often tries to reduce the number of seeding locations, i.e. locations from where a clone migrated to another location, and as a result, may miss plausible migration histories in many cases. For Ovarian Cancer 4, both Metient and MACHINA inferred only two solutions, one for considering either ovary as the primary location, where migration only happened from the primary location to the other sites. Thus, Metient and MACHINA overlook the possibility of metastasis-to-metastasis spread, which is frequent in ovarian cancer due to the peritoneal cavity lacking distinct physical barriers. Nevertheless, one may wish to consider the seeding clone criterion for certain cancer types. This is supported by MACH2 as an additional filtering step post inference. In Fig. S9b, for Prostate Cancer A24, prioritizing solutions with the fewest seeding locations yielded two solutions with four shared migrations. While MACH2-UMC assigned a $6/8 = 0.75$ probability to the migration from the right axillary lymph node to the right diaphragm, minimizing seeding locations increased this probability to 1. On the other hand, 3 migrations from xiphoid and right rib did not appear in any of the solutions after prioritization.

5.2.2 MACH2 Inferred Solutions Are Consistent with Orthogonal Experimental Data

We found that solutions inferred by MACH2 in UMC mode that MACHINA and Metient did not infer better match orthogonal experimental data. We present two such examples here. First, for non-small-cell Lung Cancer CRUK0063 [39], samples were collected at three time points: (i) on day 0, a primary lung sample (P) and a metastatic lymph node sample (LN); (ii) on day 467, after relapse, a paravertebral sample (R); and (iii) on day 857, post-mortem, four lung samples (L1-L4) and one from the paravertebral area (B). We used the clonal tree T and observed labeling $\hat{\ell}$ reported in the original paper, which were inferred by CONIPHER [12] and contained $|V(T)| = 46$ clones (Fig. 4a). Running MACH2-UMC, which does not consider sample collection time, produced 52 solutions with 4 unobserved clones, 16 migrations, and 10 comigrations (Fig. 4b). Strikingly, for all the 52 solutions, the initial seeding of the relapse sample R (collected at day 467) was from the primary tumor P, involving a migration from Clone 20 (labeled by P) to Clone 10 (labeled by R). All other seeding events involving the samples collected post mortem (on day 857) corresponded to migrations of clones descendant from Clone 10.

On the other hand, the three solutions inferred by Metient, which included the MACHINA solution, had more unobserved clones (ranging from 6 to 8) but fewer migrations (ranging from 12 to 13) and the same number of 10 comigrations (Fig. 4c). Importantly, in these solutions, P (collected at day 0) seeded L2 (collected post mortem at day 857), which in turn, seeded R (collected at day 467). This is an unrealistic explanation, as it would imply the existence of undetected tumor in the L2 region at or prior to the collection time of the relapse (R) sample. Closer inspection reveals that the three Metient/MACHINA solutions designate L2 as the origin of Clones 10 and 45, leading to a single migration into L2 at the cost of assigning Clone 45 an unobserved location. On the other hand, as MACH2-UMC prioritizes the unobserved clones criterion, it accepts more migrations into L2 (3 migrations) since it leads to fewer unobserved clones, including Clone 45 which is labeled solely by its single observed location R.

Second, for Prostate Cancer A10 [40] with $|V(T)| = 8$ clones sequenced in $|\Sigma| = 4$ locations (shown in Fig. 4d with the mapping to original clone labels provided in Table S2), MACH2-UMC inferred a solution space composed of a single refinement with 1 unobserved clone, 3 migrations, and 3 comigrations (Fig. 4e). In contrast, both Metient and MACHINA yielded the exact same refinement with 2 unobserved clones and the same number of migrations and

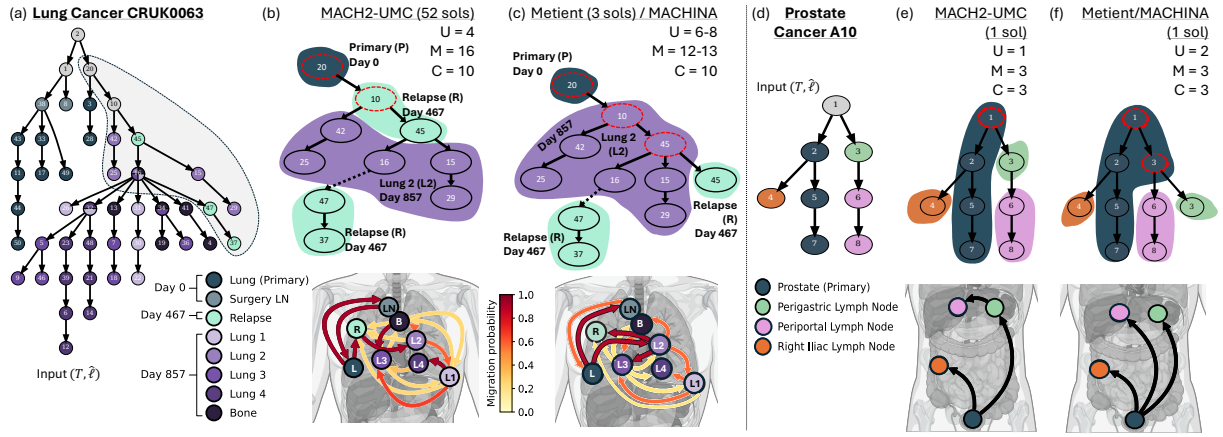


Figure 4: **MACH2 solutions match orthogonal experimental data.** (a) Clonal tree T with observed labeling $\hat{\ell}$ (colors) for Lung Cancer CRUK0063 [39]. (b) Summary graph and subtree present in all 52 MACH2-UMC solutions with colors and contours indicating locations, red dashed nodes indicating unobserved clones, and dashed edge indicating a path in the actual refinements. In all MACH2 solutions, Relapse (R) was seeded initially from the Primary Lung (L), agreeing with sample collection times. (c) Summary graph and subtree present in all 3 Metient solutions. In these solutions, Relapse (R) was seeded from post-mortem sample Lung 2 (L2), disagreeing with sample collection times. Additionally, unlike MACH2 solutions, Clone 45 is unobserved (red dashed) in L2 in the Metient solutions. (d) Clonal tree T with observed labeling $\hat{\ell}$ (colors) for Prostate Cancer A10. (e) In the unique MACH2-UMC solution, Clone 3 is observed only in the periportal lymph node, leading to seeding of the periportal lymph node from the perigastric lymph node. (f) Metient and MACHINA additionally assign Clone 3 to the prostate as an unobserved clone (red dashed), resulting in metastasis from the primary tumor only. U: unobserved clones, M: migrations, and C: comigrations.

comigrations as the MACH2 refinement (Fig. 4f). MACH2-UMC had one fewer unobserved clone as it assigned Clone 3 to the sole location where it was observed, the perigastric lymph node, suggesting a migration from the perigastric lymph node to the periportal lymph node. Since Metient and MACHINA do not attempt to minimize the number of unobserved clones but instead seek to minimize the number of seeding locations (in addition to migrations and comigrations), these methods suggest that Clone 3 originated in the prostate and then migrated to the periportal lymph node. Consequently, all metastases were seeded from the primary tumor at the expense of an additional unobserved clone. Importantly, MACH2's location assignment of Clone 3 matched experimental findings [40]. While the input clonal tree was constructed from whole-genome sequencing data (at a coverage of $55\times$), additional targeted deep resequencing (with a coverage of $471\times$) failed to detect Clone 3 in the prostate. This combined with the physical proximity of the perigastric and the periportal lymph nodes and the fact that the two periportal lymph node clones (Clones 6 and 8) are descendants of Clone 3 suggests that the MACH2 migration history is more plausible than the history inferred by both Metient and MACHINA.

6 Discussion

In this study, we introduced MACH2, a new framework for reconstructing metastatic cancer migration histories. Unlike previous methods, MACH2 additionally minimizes the number of unobserved clones, accepts flexible criteria ordering as input, and returns all optimal solutions instead of a single or limited subset of solutions. Moreover, our definition of tree refinement eliminates the need to add leaves or near-duplicate solutions (Fig. S11). To support efficient solution enumeration, we have proposed a compact representation of refinements and demonstrated a bijection between refinements and their compact forms, alongside proving complexity for various criteria orderings. Experiments on simulated data showed MACH2 in UMC mode as the best performer, while on real data, MACH2-UMC revealed biologically meaningful migration histories. There are several future directions. First, we plan to further experiments to determine under which conditions specific criteria orderings are preferred. Second, we plan to support organotropism data, as used in Metient. Third, we plan to account for mutational signature shifts in metastatic samples [44, 45] to further reduce uncertainty in migration history inference. Fourth, providing Metient with MACH2 solutions, which are guaranteed to be complete and optimal, might improve Metient's performance. Finally, we plan to apply MACH2 to large cohorts of metastatic cancers to gain deeper insights into the migration trends specific to each primary cancer type as well as study whether repeated migration trajectories can be used to reduce uncertainty.

References

[1] Christine L Chaffer and Robert A Weinberg. A perspective on cancer cell metastasis. *science*, 331(6024):1559–1564, 2011.

[2] Karuna Ganesh and Joan Massague. Targeting metastatic cancer. *Nature medicine*, 27(1):34–44, 2021.

[3] Elizabeth Comen, Larry Norton, and Joan Massague. Clinical implications of cancer self-seeding. *Nature Reviews Clinical Oncology*, 8(6):369–377, 2011.

[4] Doris P Tabassum and Kornelia Polyak. Tumorigenesis: it takes a village. *Nature Reviews Cancer*, 15(8):473–483, 2015.

[5] J Zachary Sanborn, Jongsuk Chung, Elizabeth Purdom, Nicholas J Wang, Hojabr Kakavand, James S Wilmott, Timothy Butler, John F Thompson, Graham J Mann, Lauren E Haydu, et al. Phylogenetic analyses of melanoma reveal complex patterns of metastatic dissemination. *Proceedings of the National Academy of Sciences*, 112(35):10995–11000, 2015.

[6] Mark B Faries, Shawn Steen, Xing Ye, Myung Sim, and Donald L Morton. Late recurrence in melanoma: clinical implications of lost dormancy. *Journal of the American College of Surgeons*, 217(1):27–34, 2013.

[7] Jason A Somarelli, Kathryn E Ware, Rumen Kostadinov, Jeffrey M Robinson, Hakima Amri, Mones Abu-Asab, Nicolaas Fourie, Rui Diogo, David Swofford, and Jeffrey P Townsend. Phylooncology: Understanding cancer through phylogenetic analysis. *Biochimica et Biophysica Acta (BBA)-Reviews on Cancer*, 1867(2):101–108, 2017.

[8] Q Wei, Z Ye, X Zhong, L Li, C Wang, RE Myers, JP Palazzo, D Fortuna, A Yan, SA Waldman, et al. Multi-region whole-exome sequencing of matched primary and metastatic tumors revealed genomic heterogeneity and suggested polyclonal seeding in colorectal cancer metastasis. *Annals of oncology*, 28(9):2135–2141, 2017.

[9] Zheng Hu, Jie Ding, Zhicheng Ma, Ruping Sun, Jose A Seoane, J Scott Shaffer, Carlos J Suarez, Anna S Berghoff, Chiara Cremolini, Alfredo Falcone, et al. Quantitative evidence for early metastatic seeding in colorectal cancer. *Nature genetics*, 51(7):1113–1122, 2019.

[10] Zheng Hu, Zan Li, Zhicheng Ma, and Christina Curtis. Multi-cancer analysis of clonality and the timing of systemic spread in paired primary tumors and metastases. *Nature genetics*, 52(7):701–708, 2020.

[11] David Brown, Dominiek Smeets, Borbála Székely, Denis Larsimont, A Marcell Szász, Pierre-Yves Adnet, Françoise Rothé, Ghizlane Rouas, Zsófia I Nagy, Zsófia Faragó, et al. Phylogenetic analysis of metastatic progression in breast cancer using somatic mutations and copy number aberrations. *Nature communications*, 8(1):14944, 2017.

[12] Kristiana Grigoriadis, Ariana Huebner, Abigail Bunkum, Emma Colliver, Alexander M Frankell, Mark S Hill, Kerstin Thol, Nicolai J Birkbak, Charles Swanton, Simone Zaccaria, et al. CONIPHER: a computational framework for scalable phylogenetic reconstruction with error correction. *Nature Protocols*, pages 1–25, 2023.

[13] Leah L Weber, Chuanyi Zhang, Idoia Ochoa, and Mohammed El-Kebir. Phertilizer: Growing a clonal tree from ultra-low coverage single-cell dna sequencing of tumors. *PLoS computational biology*, 19(10):e1011544, 2023.

[14] Mohammed El-Kebir, Gryte Satas, Layla Oesper, and Benjamin J Raphael. Inferring the mutational history of a tumor using multi-state perfect phylogeny mixtures. *Cell systems*, 3(1):43–53, 2016.

[15] Hamim Zafar, Anthony Tzen, Nicholas Navin, Ken Chen, and Luay Nakhleh. Sifit: inferring tumor trees from single-cell sequencing data under finite-sites models. *Genome biology*, 18:1–20, 2017.

[16] Leah L Weber, Anna Hart, Idoia Ochoa, and Mohammed El-Kebir. Pharming: Joint clonal tree reconstruction of snv and cna evolution from single-cell dna sequencing of tumors. *bioRxiv*, pages 2024–11, 2024.

[17] Jeff Wintersinger, Stephanie Dobson, John Dick, and Quaid Morris. Pairtree: fast reconstruction of cancer evolutionary history using pairwise mutation relationships. *bioRxiv*, page 372219, 2020.

[18] Salem Malikic, Andrew W McPherson, Nilgun Donmez, and Cenk S Sahinalp. Clonality inference in multiple tumor samples using phylogeny. *Bioinformatics*, 31(9):1349–1356, 2015.

[19] Mohammed El-Kebir, Gryte Satas, and Benjamin J Raphael. Inferring parsimonious migration histories for metastatic cancers. *Nature Genetics*, 50(5):718–726, 2018.

[20] Divya Koyyalagunta, Karuna Ganesh, and Quaid Morris. Inferring cancer type-specific patterns of metastatic spread. *bioRxiv*, pages 2024–07, 2024.

[21] Sudhir Kumar, Antonia Chroni, Koichiro Tamura, Maxwell Sanderford, Olumide Oladeinde, Vivian Aly, Tracy Vu, and Sayaka Miura. Pathfinder: Bayesian inference of clone migration histories in cancer. *Bioinformatics*, 36(Supplement_2):i675–i683, 2020.

[22] Montgomery Slatkin and Wayne P Maddison. A cladistic measure of gene flow inferred from the phylogenies of alleles. *Genetics*, 123(3):603–613, 1989.

[23] Andrew McPherson, Andrew Roth, Emma Laks, Tehmina Masud, Ali Bashashati, Allen W Zhang, Gavin Ha, Justina Biele, Damian Yap, Adrian Wan, et al. Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer. *Nature genetics*, 48(7):758–767, 2016.

[24] Nicola Aceto, Aditya Bardia, David T Miyamoto, Maria C Donaldson, Ben S Wittner, Joel A Spencer, Min Yu, Adam Pely, Amanda Engstrom, Huili Zhu, et al. Circulating tumor cell clusters are oligoclonal precursors of breast cancer metastasis. *Cell*, 158(5):1110–1122, 2014.

[25] Kevin J Cheung and Andrew J Ewald. A collective route to metastasis: Seeding by tumor cell clusters. *Science*, 352(6282):167–169, 2016.

[26] Kevin J Cheung, Veena Padmanaban, Vanesa Silvestri, Koen Schipper, Joshua D Cohen, Amanda N Fairchild, Michael A Gorin, James E Verdone, Kenneth J Pienta, Joel S Bader, et al. Polyclonal breast cancer metastases arise from collective dissemination of keratin 14-expressing tumor cell clusters. *Proceedings of the National Academy of Sciences*, 113(7):E854–E863, 2016.

[27] Maya Dadiani, Vyacheslav Kalchenko, Ady Yosepovich, Raanan Margalit, Yaron Hassid, Hadassa Degani, and Dalia Seger. Real-time imaging of lymphogenic metastasis in orthotopic human breast cancer. *Cancer Research*, 66(16):8037–8041, 2006.

[28] Min Yu, Aditya Bardia, Ben S Wittner, Shannon L Stott, Malgorzata E Smas, David T Ting, Steven J Isakoff, Jordan C Ciciliano, Marissa N Wells, Ajay M Shah, et al. Circulating breast tumor cells exhibit dynamic changes in epithelial and mesenchymal composition. *Science*, 339(6119):580–584, 2013.

[29] Ravikanth Maddipati and Ben Z Stanger. Pancreatic cancer metastases harbor evidence of polyclonality. *Cancer Discovery*, 5(10):1086–1097, 2015.

[30] Dena Marrinucci, Kelly Bethel, Anand Kolatkar, Madelyn S Luttgen, Michael Malchiodi, Franziska Baehring, Katharina Voigt, Daniel Lazar, Jorge Nieva, Lyudmila Bazhenova, et al. Fluid biopsy in patients with metastatic prostate, pancreatic and breast cancers. *Physical Biology*, 9(1):016003, 2012.

[31] Nicolai J Birkbak and Nicholas McGranahan. Cancer genome evolutionary trajectories in metastasis. *Cancer Cell*, 37(1):8–19, 2020.

[32] Sau Yee Kok, Hiroko Oshima, Kei Takahashi, Mizuho Nakayama, Kazuhiro Murakami, Hiroki R Ueda, Kohei Miyazono, and Masanobu Oshima. Malignant subclone drives metastasis of genetically and phenotypically heterogenous cell clusters through fibrotic niche generation. *Nature Communications*, 12(1):863, 2021.

[33] Ami Yamamoto, Andrea E Doak, and Kevin J Cheung. Orchestration of collective migration and metastasis by tumor cell clusters. *Annual Review of Pathology: Mechanisms of Disease*, 18:231–256, 2023.

[34] Mrinmoy Saha Roddur, Sagi Snir, and Mohammed El-Kebir. Enforcing temporal consistency in migration history inference. *Journal of Computational Biology*, 2024.

[35] Marie L Nydam, Kirsten B Giesbrecht, and Emily E Stephenson. Origin and dispersal history of two colonial ascidian clades in the botryllus schlosseri species complex. *PloS One*, 12(1):e0169944, 2017.

[36] David Ortiz, Stano Pekár, and Anna Bryjová. Gene flow assessment helps to distinguish strong genomic structure from speciation in an iberian ant-eating spider. *Molecular Phylogenetics and Evolution*, 180:107682, 2023.

[37] Palash Sashittal and Mohammed El-Kebir. Titus: Sampling and summarizing transmission trees with multi-strain infections. *bioRxiv*, pages 2020–03, 2020.

[38] Nicola De Maio, Colin J Worby, Daniel J Wilson, and Nicole Stoesser. Bayesian reconstruction of transmission within outbreaks using genomic variants. *PLoS computational biology*, 14(4):e1006117, 2018.

[39] Christopher Abbosh, Nicolai J Birkbak, Gareth A Wilson, Mariam Jamal-Hanjani, Tudor Constantin, Raheleh Salari, John Le Quesne, David A Moore, Selvaraju Veeriah, Rachel Rosenthal, et al. Phylogenetic ctdna analysis depicts early-stage lung cancer evolution. *Nature*, 545(7655):446–451, 2017.

[40] Gunes Gundem, Peter Van Loo, Barbara Kremeyer, Ludmil B Alexandrov, Jose MC Tubio, Elli Papaemmanuil, Daniel S Brewer, Heini ML Kallio, Gunilla Högnäs, Matti Annala, et al. The evolutionary history of lethal metastatic prostate cancer. *Nature*, 520(7547):353–357, 2015.

[41] Mohammed El-Kebir. Parsimonious migration history problem: complexity and algorithms. In *18th International Workshop on Algorithms in Bioinformatics (WABI 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[42] Kaustubh Datta, Michael Muders, Heyu Zhang, and Donald J Tindall. Mechanism of lymph node metastasis in prostate cancer. *Future oncology*, 6(5):823–836, 2010.

[43] Chao Chen, Huadong He, Zhijian Yu, Yuansong Qiu, and Xuliang Wang. Renal and retroperitoneal metastasis from prostate adenocarcinoma: a case report. *World Journal of Surgical Oncology*, 14:1–3, 2016.

[44] Charles W Ashley, Arnaud Da Cruz Paula, Rahul Kumar, Diana Mandelker, Xin Pei, Nadeem Riaz, Jorge S Reis-Filho, and Britta Weigelt. Analysis of mutational signatures in primary and metastatic endometrial cancer reveals distinct patterns of dna repair defects and shifts during tumor progression. *Gynecologic oncology*, 152(1):11–19, 2019.

[45] Lindsay Angus, Marcel Smid, Saskia M Wilting, Job van Riet, Arne Van Hoeck, Luan Nguyen, Serena Nik-Zainal, Tessa G Steenbruggen, Vivianne CG Tjan-Heijnen, Mariette Labots, et al. The genomic landscape of metastatic breast cancer highlights changes in mutation and signature frequencies. *Nature genetics*, 51(10):1450–1458, 2019.

[46] Richard M Karp. On the computational complexity of combinatorial problems. *Networks*, 5(1):45–68, 1975.

[47] Leslie G Valiant. The complexity of enumeration and reliability problems. *siam Journal on Computing*, 8(3):410–421, 1979.

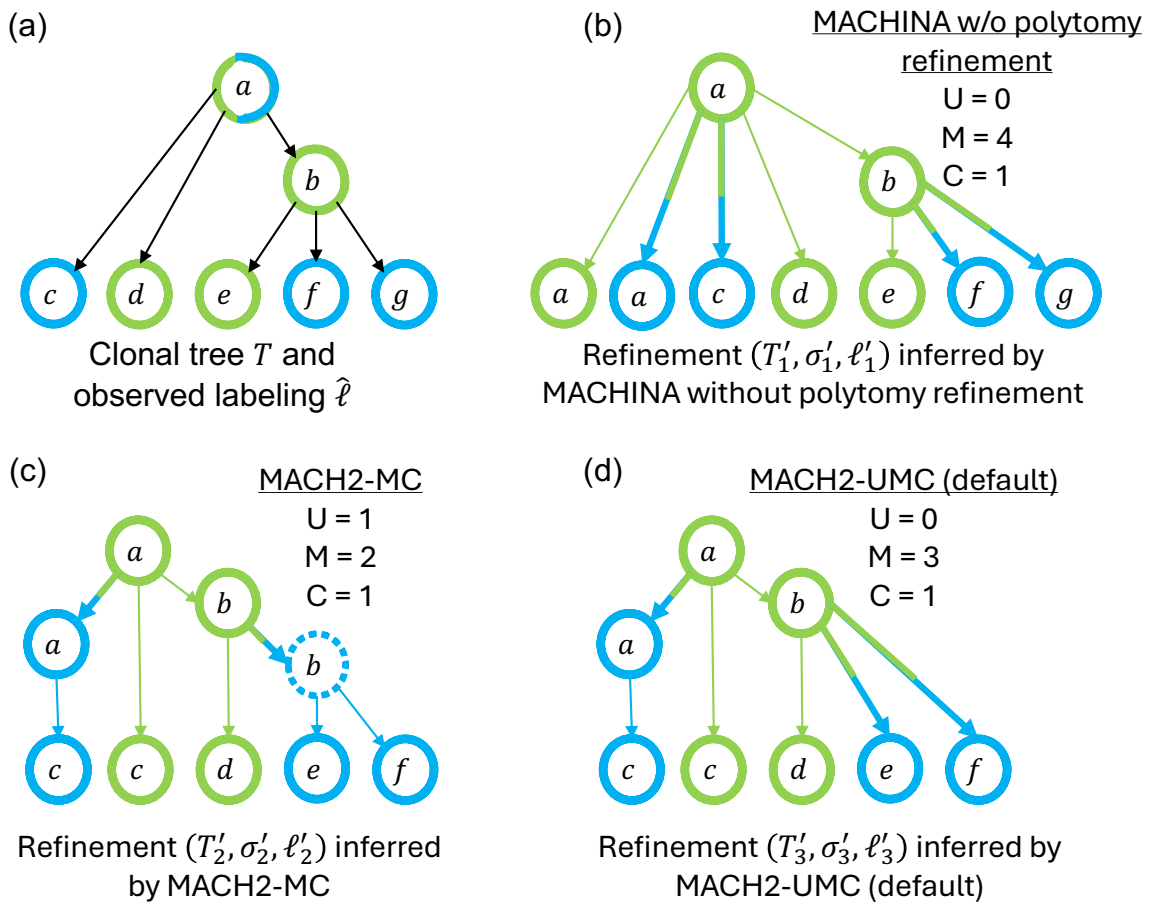


Figure S1: **Minimizing the number of unobserved clones led to more realistic scenarios supported by the data.** (a) Input clonal tree T with observed labeling $\hat{\ell}$ (colors). (b) MACHINA without polytomy refinement returns a solution $(T_1', \sigma_1', \ell_1')$ with 4 migrations and without any unobserved clones. Instead of refining node a, which is observed in two locations, MACHINA without polytomy refinement pulls down two leaves from a and labels them with the locations where a is observed. (c) MACH2-MC (also MACHINA with polytomy refinement) aggressively minimizes the number of migrations by refining nodes a and b, resulting in solution $(T_2', \sigma_2', \ell_2')$ with 2 migrations. Since b is present only in the green location, refining b produces an unobserved clone (dashed) labeled with the blue location, which is not supported by the input data. (d) MACH2-UMC (default) returns solution $(T_3', \sigma_3', \ell_3')$, where only the node a observed in two locations is refined. Thus, compared to $(T_2', \sigma_2', \ell_2')$, the solution $(T_3', \sigma_3', \ell_3')$ achieves zero unobserved clones at the expense of only one additional migration.

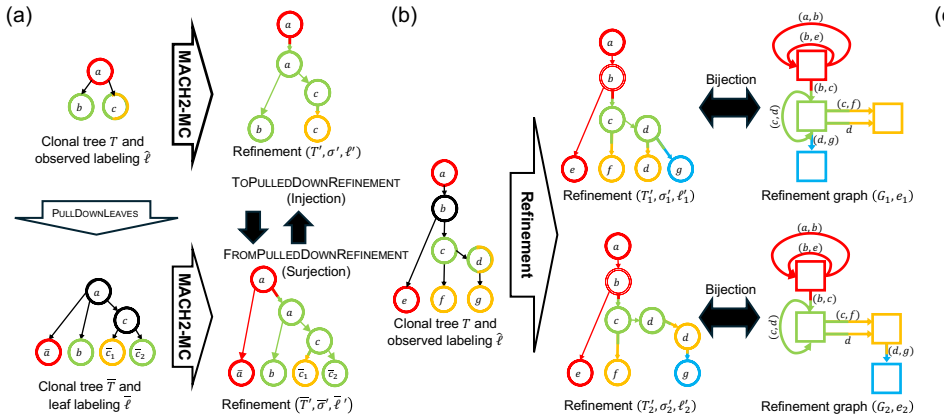
A Additional Figures and Tables

B Additional Proofs from Combinatorial Characterization and Complexity

B.1 Preliminaries

We denote the set of unobserved clones, migrations and comigrations by $\mathcal{U}(T', \sigma', \ell')$, $\mathcal{M}(T', \sigma', \ell')$ and $\mathcal{C}(T', \sigma', \ell')$, respectively, to distinguish them from their respective cardinalities $U(T', \sigma', \ell')$, $M(T', \sigma', \ell')$ and $C(T', \sigma', \ell')$.

Let T'_u indicate the subgraph of T' induced by nodes $u' \in v(T')$ such that $\sigma'(u') = u$ (Definition 2). By condition (iii) of Problem 1 it holds that for each refinement (T', ℓ', σ'), two distinct nodes $u' \neq u''$ of T' that correspond to the same node $u = \sigma'(u') = \sigma'(u'')$ of T must be labeled by distinct locations, i.e. $\ell'(u') \neq \ell'(u'')$. The notation u'_s makes use of this fact to refer to the node u' of T such that $\sigma'(u') = u$ and $\ell'(u') = s$ (if such node exists). We denote the subtree of T rooted at u by $T[u]$.



(c)

$\mathcal{F}$	DEC-PMH-TR	#PMH-TR
M	Open	Open
MU	Open	Open
MC	NP-complete [here]	Open
MUC	NP-complete [here]	#P-complete [here]
MCU	NP-complete [here]	#P-complete [here]
U	P [here]	Open
UM	Open	Open
UC	Open	Open
UMC	NP-complete [here]	#P-complete [here]
UCM	NP-complete [here]	Open
C	P [El-Kebir et al. 2018]	Open
CM	NP-complete [El-Kebir 2018]	Open
CU	Open	Open
CMU	NP-complete [here]	Open
CUM	Open	Open

Figure S2: **Combinatorial characterization of PMH-TR.** (a) Pulling down leaves, as done by MACHINA [19] and Metient [20] maintains optimality of the original solution space when $\mathcal{F} \in \{\mathrm{MC}, \mathrm{CM}\}$. (b) MACH2 relies on a compact and equivalent characterization of solutions as refinement graphs. We prove there exists a bijection between refinements (T', σ', ℓ') and edge-labeled refinement graphs (R, e) of a given tree T with observed labeling $\hat{\ell}$. Here, node labels indicate mapping functions σ_1' and σ_2', and colored nodes are used to illustrate the observed labeling in $(T, \hat{\ell})$, and the location labeling in $(T_1', \sigma_1', \ell_1')$ and $(T_2', \sigma_2', \ell_2')$. Nodes colored black in $(T, \hat{\ell})$ depict that the node was not observed in any location, and double-edged nodes in $(T_1', \sigma_1', \ell_1')$ and $(T_2', \sigma_2', \ell_2')$ represent unobserved clones. (c) Prior and new complexity results established in this paper.

Dataset	Patient ID	#Nodes	#Locations	#Solutions (MACH2-UMC)	#Unobserved Clones (MACH2-UMC)	#Migrations (MACH2-UMC)	#Comigrations (MACH2-UMC)	#Solutions (Metient)	#Unobserved Clones (Metient)	#Migrations (Metient)	#Comigrations (Metient)	#Solutions (MACHINA)	#Unobserved Clones (MACHINA)	#Migrations (MACHINA)	#Comigrations (MACHINA)
Lung cancer [39]	CRUK0063	46	8	52	4	16	10	5	6-12	12-20	7-10	1	8	12	7
Prostate cancer [40]	A10	8	4	1	1	3	3	2	2-3	3	3	1	2	3	3
	A12	5	4	6	1	3	3	1	2	3	3	1	2	3	3
	A17	13	6	1	1	9	7	5	3-8	7-9	5	1	4	7	5
	A21	15	7	11	2	6	6	8	4-7	6	6	1	4	6	6
	A22	16	10	4	0	26	16	8	0-10	30-38	10-23	1	0	26	16
	A24	10	5	8	0	9	5	3	0-5	9-13	4-6	1	1	9	5
	A29	5	2	1	1	1	1	2	1-2	1	1	1	1	1	1
	A31	10	5	4	1	10	4	3	1-3	10-12	4-7	1	1	10	4
	A32	12	5	2	0	9	7	1	2	8	4	1	3	7	5
	A34	14	3	3	3	2	2	1	3	2	2	1	3	2	2
Ovarian cancer [23]	2	6	2	1	1	1	1	1	1	1	1	1	1	1	1
	10	6	3	1	0	6	2	1	0	6	2	1	0	6	2
	1	9	7	44	0	12	7	8	1-3	11-13	6-9	1	1	11	7
	3	7	8	8	0	25	11	7	0-3	25-27	7-17	2	0	25	11
	4	8	4	32	0	6	3	2	0	6	3	2	0	6	3
	7	5	6	8	0	10	5	8	0-3	11-12	5-7	2	1-2	11	6
	9	3	3	6	0	4	2	3	0-1	4-5	2	2	0	4	2

Table S1: **Detailed real data results.** For each real data instance $(T, \hat{\ell})$, we list the number $|V(T)|$ of nodes, the number $|\Sigma|$ of locations, the number of solutions (T', σ', ℓ'), the number $\mathrm{U}(T', \sigma', \ell')$ of unobserved clones, the number $M(T', \sigma', \ell')$ of migrations and the number $C(T', \sigma', \ell')$ of comigrations, identified by MACH2-UMC, Metient [20], and MACHINA [19].

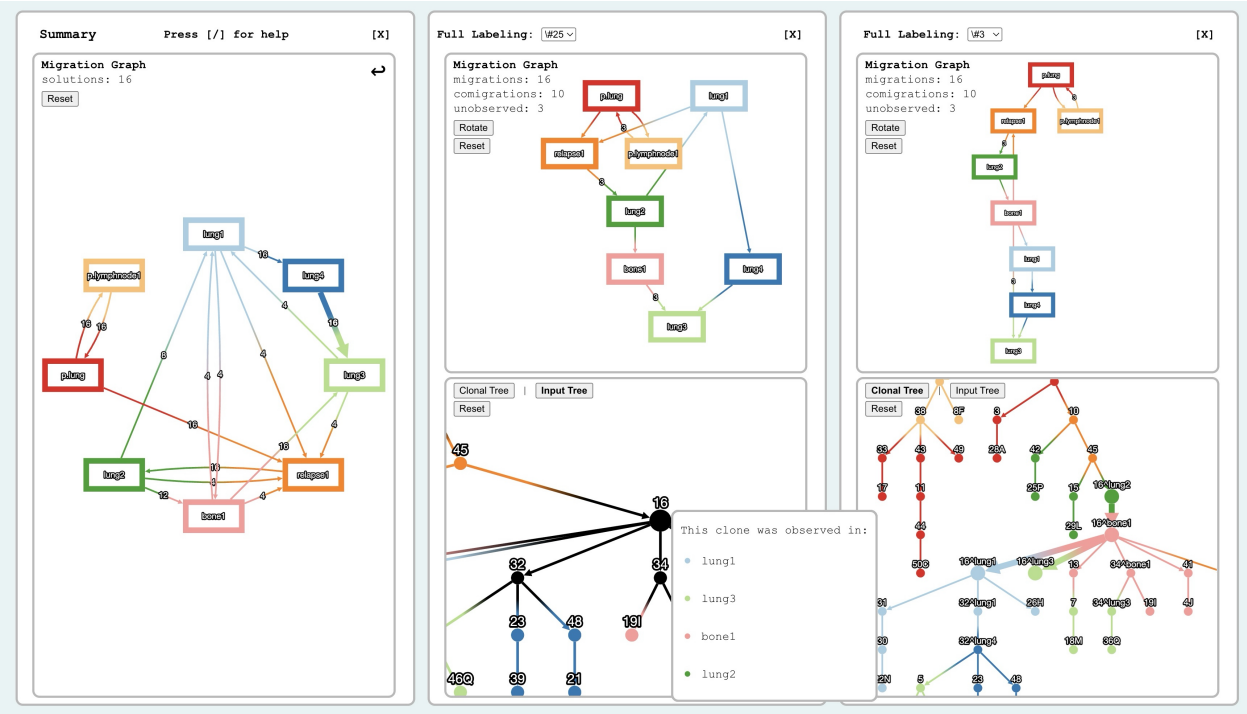


Figure S3: Three panel summary view in `mach2-viz` of the TracerX lung cancer study. The left panel shows the summary graph, computed through the union of edges across the solution space. There is an enforcement of the migration between `lung4` and `lung3`. The right two panels shows a two solutions with a different migration graph each. The middle panel shows an input clonal tree with multiple observed locations for an unlabeled node being hovered over. The corresponding polytomy resolution is highlighted in the rightmost panel's output clonal tree.

Clone name in this manuscript	Clone name in Gundem et al. [40]
1	grey
2	dark pink
3	gold
4	dark brown
5	pink
6	dark purple
7	blue
8	orange

Table S2: **Mapping of clone names for Prostate Cancer A10.**

B.2 Additional Properties of the Solution Space

We have the following lower bound on the number $\mathrm{U}(T', \sigma', \ell')$ of unobserved clones.

Proposition 5. *The number $\mathrm{U}(T', \sigma', \ell')$ of unobserved clones is at least the number $|\{v \in V(T) \mid \hat{\ell}(v) = \emptyset\}|$ of clones not observed anywhere.*

The following previously-proven proposition establishes a lower bound on the number $M(T', \sigma', \ell')$ of migrations in PMH-TR solution (T', σ', ℓ').

Proposition 6 (El-Kebir et al. [19])**.** *For any refinement (T', σ', ℓ') of $(T, \hat{\ell})$, the number $M(T', \sigma', \ell')$ of migrations is at least $|\Sigma| - 1$, where $\Sigma = \bigcup_{v \in V(T)} \hat{\ell}(v)$ is the set observed locations.*

Here, we provide the necessary and sufficient conditions to meeting this lower bound.

Proposition 7. *It holds that $M(T', \sigma', \ell') = |\Sigma| - 1 = |\bigcup_{v \in V(T)} \hat{\ell}(v)| - 1$ if and only if (i) $\ell'(v') \in \Sigma$ for each node v' of T' and (ii) for any two nodes w', w'' of T' where $\ell'(w') = \ell'(w'')$, each node in the undirected path from w' to w'' is labeled with $\ell'(w') = \ell'(w'')$.*

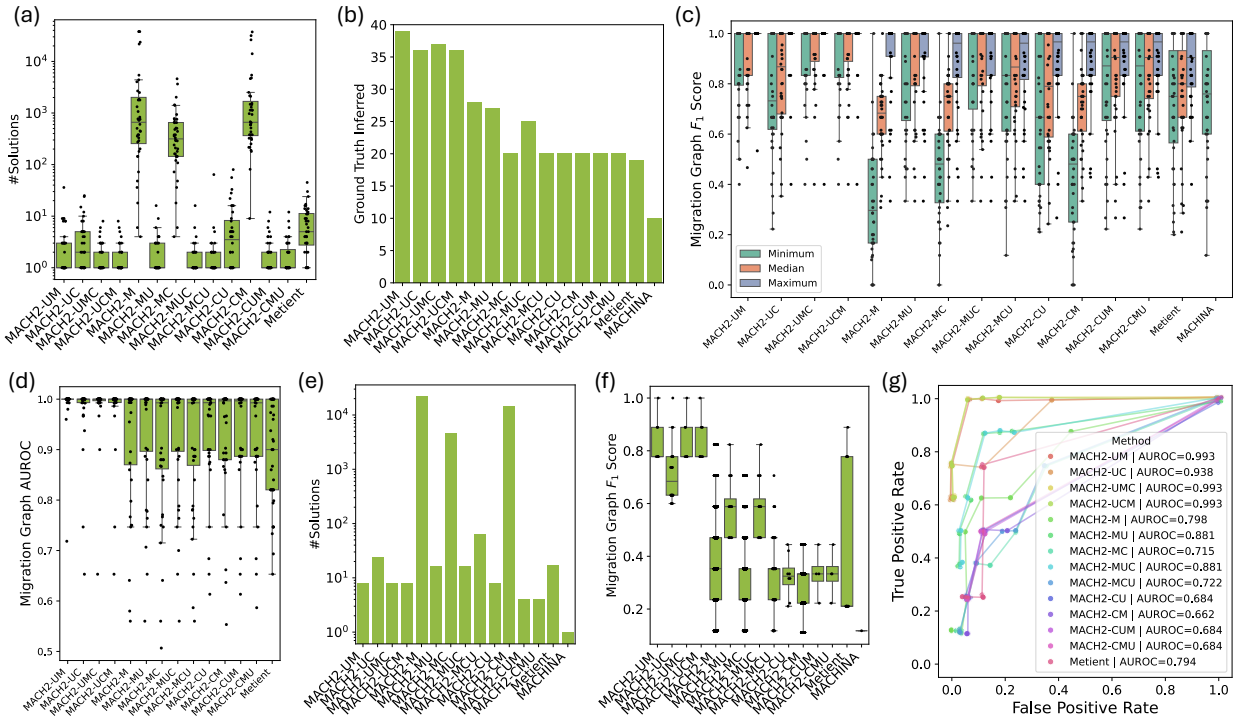


Figure S4: **Comparison between MACH2 with 13 criteria orderings, Metient [20] and MACHINA [19] on simulated data.** (a) MACH2 with 13 criteria orderings and Metient inferred multiple solutions for the majority of the instances. (b) Number of times each method included the ground-truth solution. (c) Minimum, median, and maximum migration graph F_1 score across all instances. (d) Distribution of AUROC values across all instances. (e) The number of identified solutions of the solution spaces inferred by different methods for the instance discussed in Section 5.1 and shown in Fig. 2a. (f) ROC curves obtained from summary graphs of the discussed simulation instance. (g) Migration graph F_1 score of the discussed simulation instance.

The next proposition gives an upper bound for the number $C(T', \sigma', \ell')$ of comigrations.

Proposition 8. *It holds that* $C(T', \sigma', \ell') \leq |E(R)| \leq |E(T)| + |V(T)|(|\Sigma| - 1)$.

Finally, we have the following propositions that apply to specific criteria orderings $\mathcal{F}$, bounding the number $|V(T'_u)|$ of refined nodes $u' \in V(T')$ that map to original node u, i.e. $\sigma'(u') = u$.

Proposition 9. *Let* (T', σ', ℓ') *be a refinement that minimizes the number of unobserved clones first. Then, for any node* $u \in V(T)$ *observed in less than two locations, i.e.* $|\hat{\ell}(u)| < 2$*, it holds that* $|V(T'_u)| = 1$.

Proposition 10. *Let* (T', σ', ℓ') *be a refinement that minimizes the number of migrations first. Then, there exists an optimal refinement* (T', σ', ℓ') *such that for any node* $u \in V(T)$ *without polytomies and observed in less than two locations, i.e.* $|\delta_T(u)| < 2$ *and* $|\hat{\ell}(u)| < 2$*, it holds that* $|V(T'_u)| = 1$.

B.3 Combinatorial Characterization

For completeness, we give the pseudocode for the procedure PULLDOWNLEAVES$(T, \hat{\ell})$ in Algorithm S1.

In this section, we prove the following proposition.

(Main text) Proposition 1. *Let* $(\overline{T}, \overline{\ell}) =$ PULLDOWNLEAVES$(T, \hat{\ell})$ *and* $\mathcal{F} \in \{\text{MC}, \text{CM}\}$*. Then, the solution space* PMH-TR$(\mathcal{F}, T, \hat{\ell})$ *is a subset of the solution space* FROMPULLEDDOWNREFINEMENT(PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$).

Let $\mathcal{X}(T, \hat{\ell})$ be the set composed of all observed node-location pairs (v, s) where $v \in V(T)$ and $s \in \hat{\ell}(v)$. By definition, PULLDOWNLEAVES induces a bijection $\kappa : \mathcal{X}(T, \hat{\ell}) \to L(\overline{T})$ between the observed node-location pairs $\mathcal{X}(T, \hat{\ell})$ of $(T, \hat{\ell})$ and the location-labeled leaves $L(\overline{T})$ of $(\overline{T}, \overline{\ell})$. More specifically, $\kappa(v, s) = \overline{v}$ if $\overline{v}$ is a pulled-down

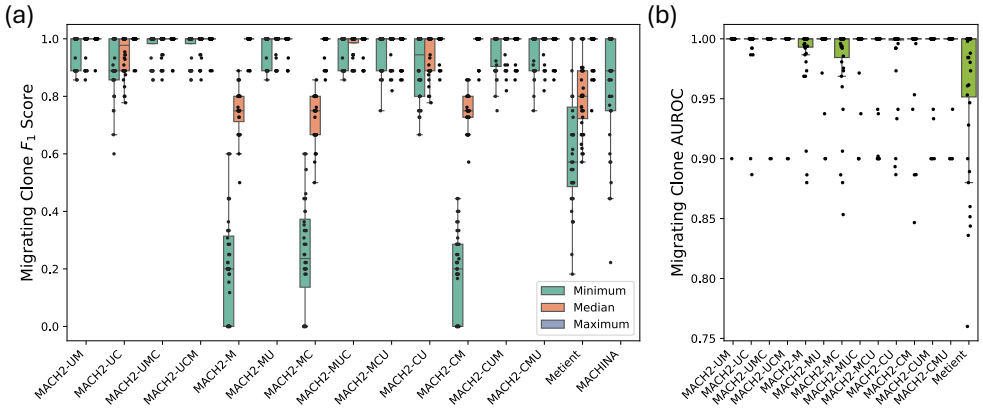


Figure S5: (a) Minimum, median, and maximum migrating clone F_1 score (Section E), and (b) migrating clone area under receiver operating characteristics curve (AUROC) for MACH2 with 13 criteria orderings, Metient, and MACHINA.

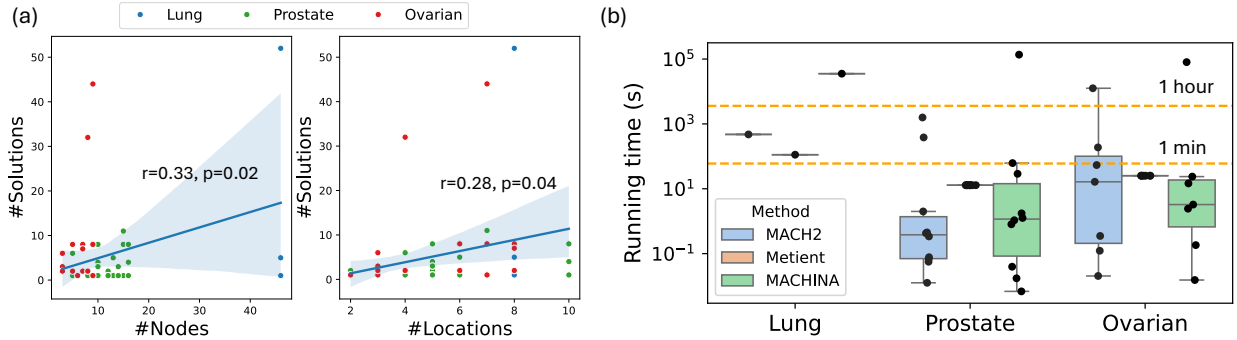


Figure S6: (a) The number of solutions for MACH2-UMC increases with the number of nodes and the number of locations. (b) MACH2 in UMC mode is slightly faster than MACHINA and MACH2. Note that Metient in calibrate mode processes an entire cohort at once. Thus we reported the mean running time per instance, calculated by dividing the total cohort time by the number of instances. All experiments were run on a server with Intel Xeon Gold 5120 dual CPUs with 14 cores each at 2.20 GHz and 512 GB RAM.

Clone name in this manuscript	Clone name in Gundem et al. [40]
1	grey
2	red
3	dark brown
4	mid blue
5	light green
6	dark purple
7	dark blue
8	orange
9	mid green
10	dark green
11	pink
12	light purple
13	gold
14	light blue
15	yellow
16	light brown

Table S3: **Mapping of clone names for Prostate Cancer A22.**

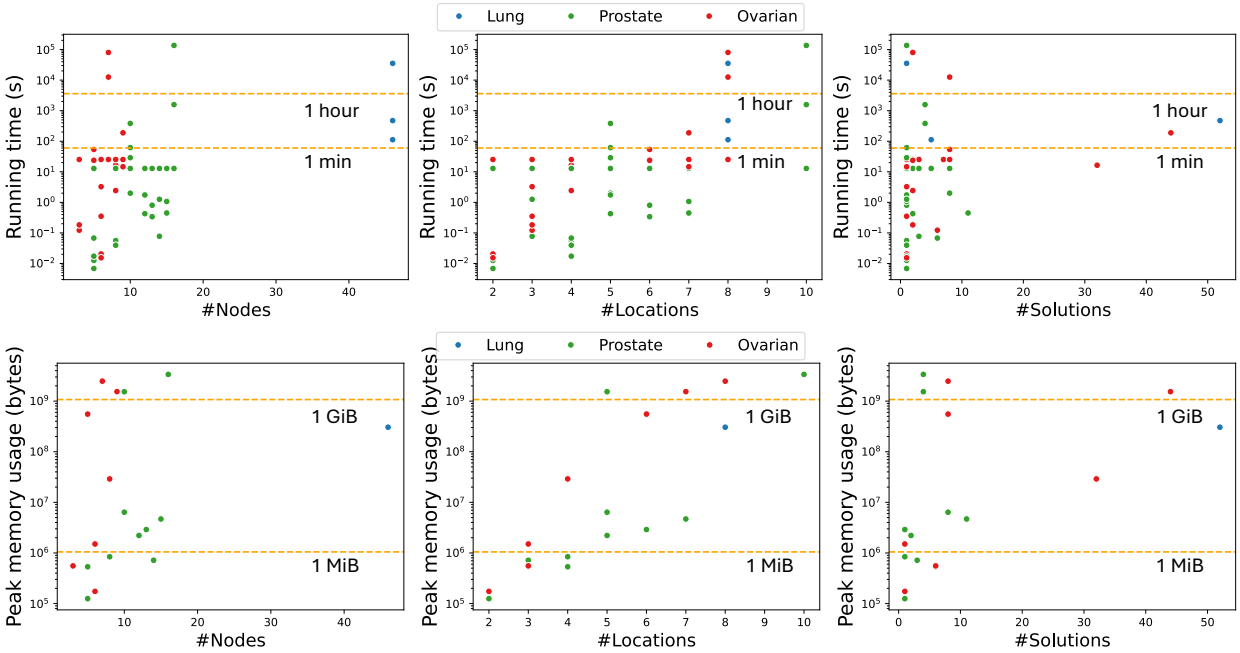


Figure S7: Running time (in seconds) and peak memory usage (in bytes) of MACH2-UUMC as functions of the number of nodes, locations, and solutions. All experiments were run on a server with Intel Xeon Gold 5120 dual CPUs with 14 cores each at 2.20 GHz and 512 GB RAM.

Algorithm S1 Construction of $(\overline{T}, \overline{\ell} : L(\overline{T}) \to \Sigma)$ by pulling down leaves from $(T, \hat{\ell} : V(T) \to 2^{\Sigma})$

procedure PULLDOWNLEAVES($T, \hat{\ell}$)
  $\overline{T} \leftarrow T$
  for each $v \in V(T)$ **do**
    Let $\overline{v} \in V(\overline{T})$ be the counterpart of v in $\overline{T}$
    if v is a leaf of T and $|\hat{\ell}(v)| = 1$ **then**
      Let $\{s\} = \hat{\ell}(v)$ be the unique location label of v
      $\overline{\ell}(\overline{v}) \leftarrow s$
    else
      for each $s \in \hat{\ell}(v)$ **do**
        $V(\overline{T}) \leftarrow V(\overline{T}) \cup \{\overline{v}'\}$
        $E(\overline{T}) \leftarrow E(\overline{T}) \cup \{(\overline{v}, \overline{v}')\}$
        $\overline{\ell}(\overline{v}') \leftarrow s$
      end for
    end if
  end for
  return $(\overline{T}, \overline{\ell})$
end procedure

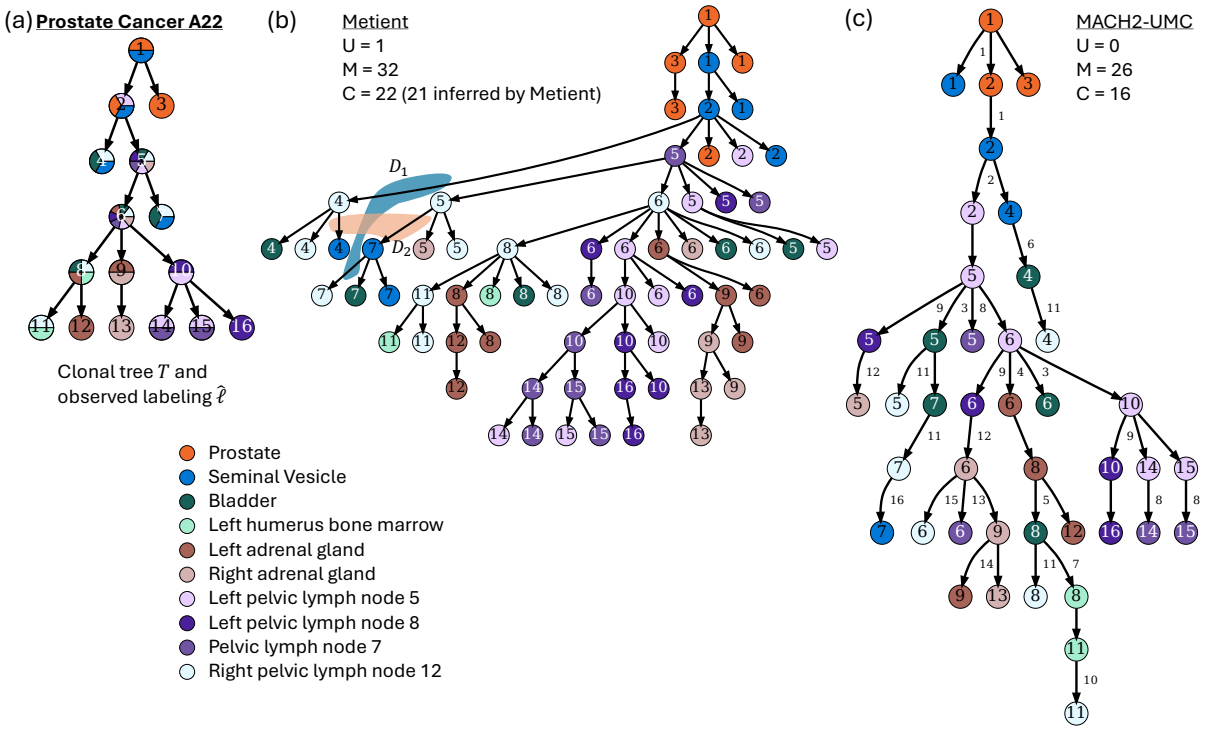


Figure S8: **MACH2 accurately infers the number of temporally consistent comigrations.** (a) Clonal tree T with observed labeling $\hat{\ell}$ for Prostate Cancer A22. (b) Metient returns a solution with 1 unobserved clone (U), 32 migrations (M) and 22 comigrations (C). But Metient reports the tree to have 21 comigrations. Metient groups the migrations indicated by the contour D_1 in one comigration and the migrations indicated by the contour D_2 in another comigration. However, this creates temporal inconsistency because none of the migrations precede one another. In other words, if migrations from the comigration D_1 happen simultaneously, then migration from the comigration D_2 cannot happen simultaneously, and vice versa. (c) MACH2 returns a solution with 0 unobserved clone (U), 26 migrations (M) and 16 temporally consistent comigrations (C), where migrations with the same label belong to the same comigration.

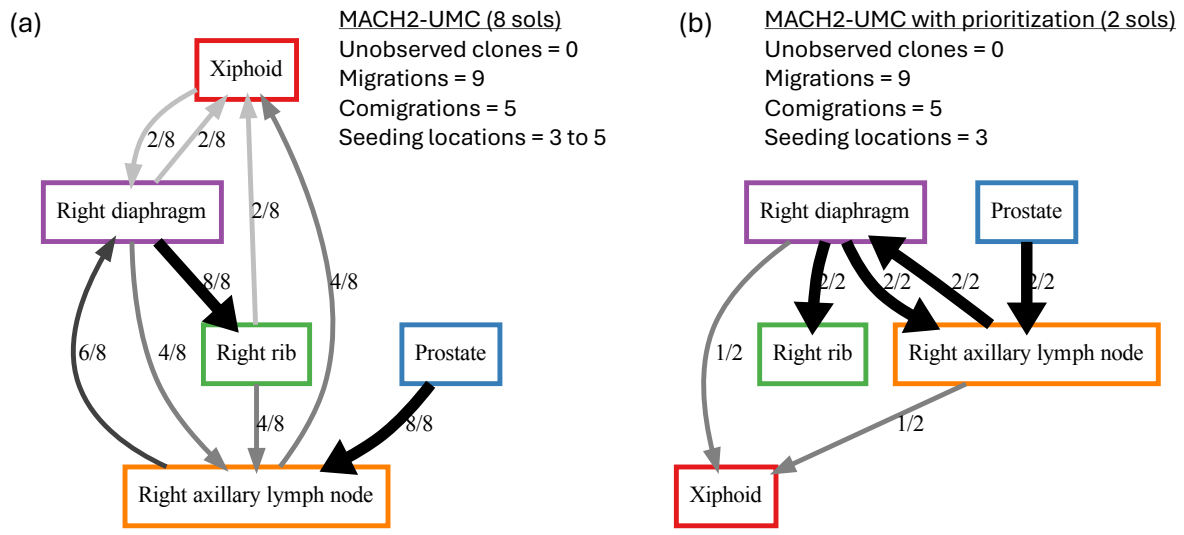


Figure S9: (a) Summary graph for Prostate Cancer A24 with 2 migrations with a probability of 1 (boldface). (b) Summary graph for the same patient after prioritizing solutions that minimizes the number of seeding locations, with 4 migrations with a probability of 1 (boldface).

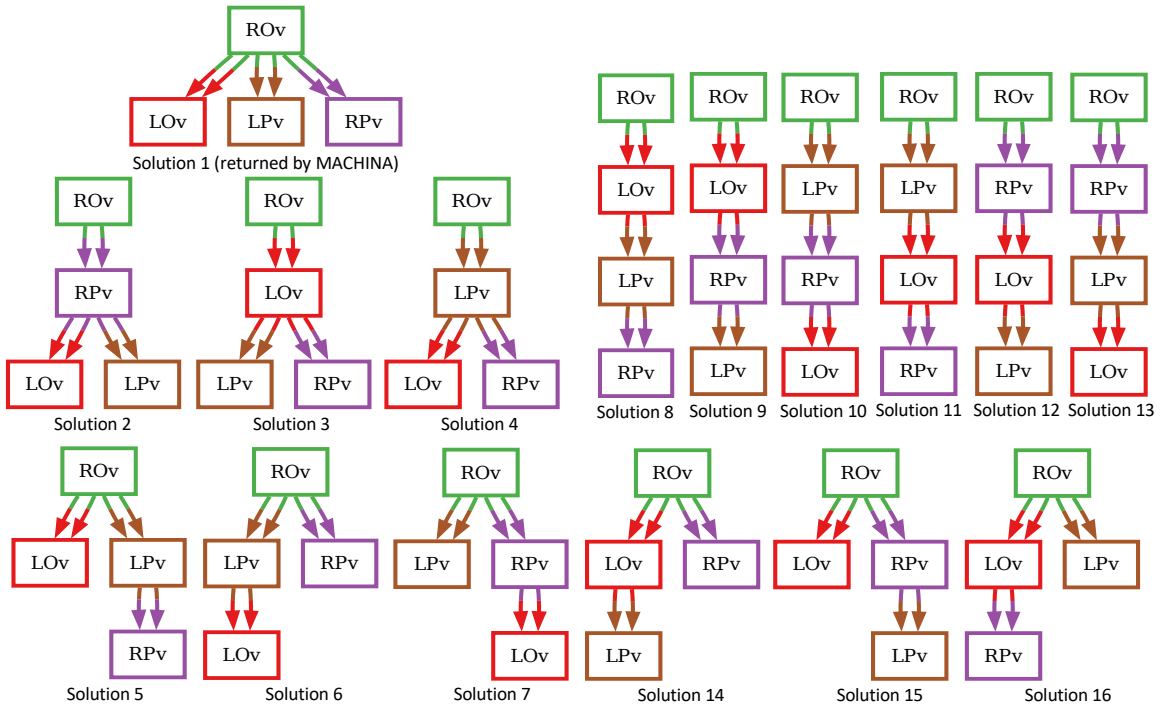


Figure S10: 16 **solutions inferred by MACH2-UMC for Ovarian Cancer** 4 **[23] when the right ovary (ROv) is considered as the primary location.** The migration graph of each solution is presented, where migration graph is the refinement graph without self-loops. Ignoring multi-edges, each migration graph admits a unique tree topology with the root labeled by the right ovary (ROv) and each of the non-roots labeled either by the left ovary (LOv), the left pelvic area (LPv), or the right pelvic area (RPv). There other 16 MACH2-UMC solutions when the left ovary is considered as the primary location can be obtained by flipping the labels of the left ovary (LOv) and the right ovary for each solution.

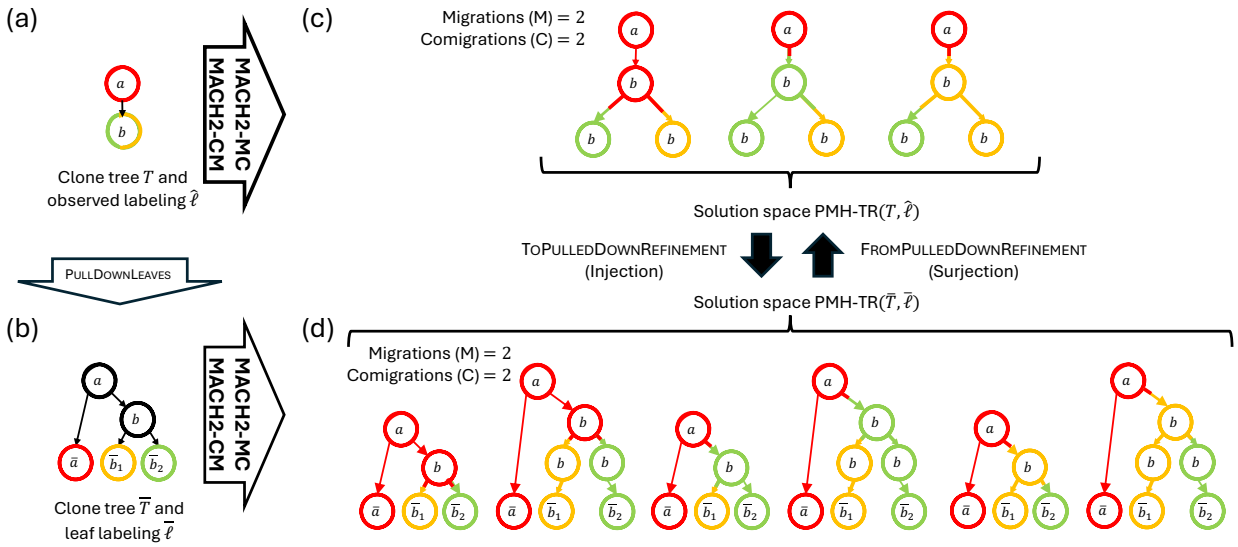


Figure S11: **Example showing that PMH-TR$(\mathcal{F}, T, \hat{\ell})$ may be a proper subset of FROMPULLEDDOWNREFINEMENT(PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$).** (a) A PMH-TR input instance $(T, \hat{\ell})$ with two clones observed in three locations. (b) The corresponding input instance $(\overline{T}, \overline{\ell})$ obtained by pulling down leaves from $(T, \hat{\ell})$. (c) The solution space PMH-TR$(\mathcal{F}, T, \hat{\ell})$ has three solutions with $\mathrm{M} = 2$ migrations and $\mathrm{C} = 2$ comigrations. (d) The solution space PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$ has three additional solutions with $\mathrm{M} = 2$ migrations and $\mathrm{C} = 2$ comigrations.

leaf, i.e. $\overline{v} \in L(\overline{T}) \setminus L(T)$ with incoming edge $(v, \overline{v})$ in $\overline{T}$ and label $\overline{\ell}(\overline{v}) = s$; and $\kappa(v, s) = v$ if v is not a pulled-down leaf of T observed in unique location s, i.e. $v \in L(T)$ and $\hat{\ell}(v) = \{s\}$. We refer to Fig. S2a for an example.

To prove Proposition 1, we first show that for any optimal solution (T', σ', ℓ') to PMH-TR instance $(\mathcal{F}, T, \hat{\ell})$, there exists a unique feasible solution $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ to PMH-TR instance $(\mathcal{F}, \overline{T}, \overline{\ell})$ with the same number of migrations and comigrations. To that end, we present a function TOPULLEDDOWNREFINEMENT to build a solution $(\overline{T}', \overline{\sigma}', \overline{\ell}') \in$ PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$ from (T', σ', ℓ'), $(T, \hat{\ell})$ and $(\overline{T}, \overline{\ell})$, with the same number of migrations and comigrations, i.e. $M(T', \sigma', \ell') = M(\overline{T}', \overline{\sigma}', \overline{\ell}')$ and $C(T', \sigma', \ell') = C(\overline{T}', \overline{\sigma}', \overline{\ell}')$. TOPULLEDDOWNREFINEMENT does this by iteratively considering all observed node-location pairs $\mathcal{X}(T, \hat{\ell})$ and modifying (T', σ', ℓ'). For each observed node-location pair $(v, s) \in \mathcal{X}(T, \hat{\ell})$, there is exactly one refined node v' of T' labeled by location $\ell'(v') = s$ that corresponds to the original node $\sigma(v') = v$ — this follows from condition (iv) of Definition 2. Moreover, $\kappa(v, s) = \overline{v}$ is the pulled down leaf from v labeled with $\overline{\ell}(\overline{v}) = s$ or node v itself if v is observed in one location. If v' is an internal node of T', we (i) add a new node $\overline{v}'$, (ii) add the edge $(v', \overline{v}')$, (iii) set the original node of new leaf $\overline{v}'$ to be $\overline{\sigma}'(\overline{v}') = \kappa(v, s) = \overline{v}$, and (iv) label with $\overline{\ell}'(\overline{v}') = s$. On the other hand, if v' is a leaf, we update the original node of v' to be $\overline{\sigma}'(v') = \kappa(v, s) = \overline{v}$. For completeness, we give the pseudocode of the procedure TOPULLEDDOWNREFINEMENT in Algorithm S2.

We denote the space of *optimal solutions* of the original PMH-TR instance by PMH-TR$(\mathcal{F}, T, \hat{\ell})$. Let $(\overline{T}, \overline{\ell}) =$ PULLDOWNLEAVES$(T, \hat{\ell})$ be the pulled down PMH-TR instance obtained from $(T, \hat{\ell})$. We denote the space of *feasible solutions* of the pulled-down instance by PMH-TR$^*(\mathcal{F}, \overline{T}, \overline{\ell})$. In our first lemma, we claim that for $\mathcal{F} \in \{\mathrm{MC}, \mathrm{MC}\}$ the function TOPULLEDDOWNREFINEMENT with domain PMH-TR$(\mathcal{F}, T, \hat{\ell})$ is an injective function for the range of feasible solutions PMH-TR$^*(\mathcal{F}, \overline{T}, \overline{\ell})$ with the same number of migrations and comigrations as solutions in PMH-TR$(\mathcal{F}, T, \hat{\ell})$.

Lemma 11. *For $\mathcal{F} \in \{\mathrm{MC}, \mathrm{MC}\}$, the function* TOPULLEDDOWNREFINEMENT *with domain* PMH-TR$(\mathcal{F}, T, \hat{\ell})$ *is an injective function for the range of feasible solutions* PMH-TR$^*(\mathcal{F}, \overline{T}, \overline{\ell})$ *with the same number of migrations and comigrations as solutions in* PMH-TR$(\mathcal{F}, T, \hat{\ell})$.

Proof. For any refinement $(T', \sigma', \ell') \in$ PMH-TR$(\mathcal{F}, T, \hat{\ell})$, the function call $(\overline{T}', \overline{\sigma}', \overline{\ell}') =$ TOPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (T', \sigma', \ell'))$ attaches edges to the existing nodes of T', sets the original node and the location labels of the new nodes, and ensures $\overline{T}$, $\overline{\sigma}'$ and $\overline{\ell}'$ are a valid tree, an original node mapping and a loca-

Algorithm S2 Construction of refinement $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ of $(\overline{T}, \overline{\ell})$ from given refinement (T', σ', ℓ') of $(T, \hat{\ell})$

procedure TOPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (T', \sigma', \ell'))$
  $(\overline{T}, \overline{\ell}) \leftarrow$ PULLDOWNLEAVES$(T, \hat{\ell})$
  $(\overline{T}', \overline{\ell}') \leftarrow (\overline{T}, \overline{\ell})$
  for each $\overline{v}' \in V(\overline{T'})$ **do**
    Let $\overline{v}$ be the node of $\overline{T}$ corresponding to $\overline{v}'$
    $\sigma(\overline{v}') \leftarrow \overline{v}$
  end for
  $X(T, \hat{\ell}) = \{(v, s) \in V(T) \times \Sigma \mid s \in \hat{\ell}(v)\}$
  for each $(v, s) \in X(T, \hat{\ell})$ **do**
    Let $v' \in V(T')$ be the unique node labeled by $\ell'(v') = s$ and $\sigma'(v') = v$
    Let $\overline{v}'$ be the node of $\overline{T}'$ corresponding to v'
    Let $\overline{v} = \kappa(v, s)$ be the leaf of $\overline{T}$ corresponding to v labeled by s
    if $v' \notin L(T')$ **then**
      $V(\overline{T}') \leftarrow V(\overline{T}') \cup \{\overline{v}''\}$
      $E(\overline{T}') \leftarrow E(\overline{T}') \cup \{(\overline{v}', \overline{v}'')\}$
      $\sigma(\overline{v}'') \leftarrow \overline{v}$
      $\overline{\ell}'(\overline{v}') \leftarrow s$
    else
      $\overline{\sigma}'(v') \leftarrow \overline{v}$
    end if
  end for
  return $(\overline{T}', \overline{\sigma}', \overline{\ell}')$
end procedure

tion labeling, respectively. Moreover, since TOPULLEDDOWNREFINEMENT only adds non-migrations to (T', σ', ℓ') to obtain $(\overline{T}', \overline{\sigma}', \overline{\ell}')$, they have identical sets of migrations and comigrations, i.e. $\mathcal{M}(T', \sigma', \ell') = \mathcal{M}(\overline{T}', \overline{\sigma}', \overline{\ell}')$ and $\mathcal{C}(T', \sigma', \ell') = \mathcal{C}(\overline{T}', \overline{\sigma}', \overline{\ell}')$, and thus identical numbers of migrations and comigrations. By Problem 1, $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ is a feasible solution to PMH-TR instance $(\mathcal{F}, \overline{T}, \overline{\ell})$ provided $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ is a refinement of $(\overline{T}, \overline{\ell})$. To that end, we show that $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ meets all the four conditions of refinement from Definition 2.

1. Condition (i) states that for any original node $\overline{v}$ in $\overline{T}$, the refined nodes $\overline{v}'$ in $\overline{T}'$ where $\overline{\sigma}'(\overline{v}') = \overline{v}$ induce a connected subtree $\overline{T}'_{\overline{v}}$ in $\overline{T}'$. There are two cases.
   - $\overline{v}$ is in T. Note that $\overline{\sigma}'(v') = \sigma'(v')$ for any refined node v' in T' that is unomodified by TOPULLEDDOWNREFINEMENT. Since T' is a refinement of T, the refined nodes v' in T' where $\sigma'(v') = \overline{v}$ induce a connected subtree $T'_{\overline{v}}$ in T'. Moreover, as TOPULLEDDOWNREFINEMENT adds or modifies leaves in T' only, a leaf in T' is also a leaf in $T'_{\overline{v}}$, and removing leaves retains the tree structure, the nodes v' in $T'_{\overline{v}}$ not modified by TOPULLEDDOWNREFINEMENT induce a subtree both in T' and $\overline{T}'$. Therefore, to prove that $\overline{T}'_{\overline{v}}$ is an induced subtree, it suffices to show that each leaf $\overline{v}'$ in $\overline{T}'_{\overline{v}}$ added or modified by TOPULLEDDOWNREFINEMENT either is connected with an unmodified node from $\overline{T}'_{\overline{v}}$, or $\overline{T}'_{\overline{v}}$ is a single-node subtree. Now two cases can arise.
     - $\overline{v}'$ is in T'. In this case, $\overline{v}'$ is the unique leaf with original node and location label being $\overline{v}$ and $\ell'(\overline{v}')$, and for node-location pair $\sigma'(\overline{v}'), \ell'(\overline{v}') = (\overline{v}, \ell'(\overline{v}')) \in \mathcal{X}$, TOPULLEDDOWNREFINEMENT modifies the corresponding original node of $\overline{v}'$ to be $\overline{\sigma}'(\overline{v}') = \kappa(\sigma'(\overline{v}'), \ell'(\overline{v}')) = \overline{v}$. By definition, $\kappa(v, s) = v$ for any original node v and location s if v is in T. Therefore, $\overline{\sigma}'(\overline{v}') = \sigma'(\overline{v}') = \overline{v}$. Thus, $\overline{v}'$ is part of subtree $T'_{\overline{u}}$ and so, either $\overline{v}'$ has an incoming edge from an unmodified node of $\overline{T}'_{\overline{u}}$, or $\overline{v}'$ is the only node in $\overline{T}'_{\overline{u}}$.
     - $\overline{v}'$ is not in T'. In this case, v' is the unique leaf with original node and location label being $\overline{v}$ and $\ell'(v')$, and for node-location pair $(\sigma'(v'), \ell'(v')) = (\overline{v}, \ell'(\overline{v}')) \in \mathcal{X}$, TOPULLEDDOWNREFINEMENT adds the

edge $(v', \overline{v}')$. Thus, $\overline{v}'$ has an incoming edge from an unmodified node v' of $\overline{T}'_{\overline{u}}$.

Therefore, $\overline{T}'_{\overline{v}}$ is a subtree.

- $\overline{v}$ is a pulled-down leaf, i.e. $\overline{v} \in L(\overline{T}) \setminus L(T)$. In this case, there is a unique node-location pair $(v, s) \in \mathcal{X}$ such that $\kappa(v, s) = \overline{v}$ by definition. Consequently, TOPULLEDDOWNREFINEMENT sets the original node of exactly one refined leaf $\overline{v}'$ in $\overline{T}'$ to $\overline{\sigma}'(\overline{v}') = \kappa(v, s) = \overline{v}$. Thus, $\overline{T}'_{\overline{v}}$ is a single-node tree.

2. Condition (ii) states that for each edge $(\overline{u}, \overline{v})$ in $\overline{T}$, there is exactly one edge $(\overline{u}', \overline{v}')$ such that $\overline{\sigma}'(\overline{u}') = \overline{u}$ and $\overline{\sigma}'(\overline{v}') = \overline{v}$. There are two cases.

   - $(\overline{u}, \overline{v})$ is in T. Since T' is a refinement of T, there is exactly one edge (u', v') in T' such that $\sigma'(u') = \overline{u}$ and $\sigma'(v') = \overline{v}$. As TOPULLEDDOWNREFINEMENT adds or modifies leaves only, and u' is an internal node, $\sigma'(u') = \overline{\sigma}'(u') = \overline{u}$. Moreover, since $T'_{\overline{v}}$ is a connected subtree, v' must be the root of $T'_{\overline{v}}$. Now, if v' is also an internal node, then $\sigma'(v') = \overline{\sigma}'(v') = \overline{v}$. On the other hand, if v' is a leaf, then $T'_{\overline{v}}$ is a single-node tree, which implies $|\hat{\ell}(\overline{v})| = 1$. Let $\hat{\ell}(\overline{v}) = \{s\}$. Then by PULLDOWNLEAVES, $\kappa(\overline{v}, s) = \overline{v}$, and so the corresponding original node in $\overline{T}'$ is $\overline{\sigma}'(v') = \overline{v}$. Note that since $\overline{T}'_{\overline{v}}$ is a connected subtree, for any edge $(\overline{u}', \overline{v}')$ in T' where $\overline{\sigma}'(\overline{v}') = \overline{v} \neq \overline{\sigma}'(\overline{u}')$, $\overline{v}'$ must be the root of $\overline{T}'_{\overline{v}}$. Since there can be only one root $r(\overline{T}'_{\overline{v}})$, there is exactly one such edge in $\overline{T}'$.
   - $(\overline{u}, \overline{v})$ is not in T. In this case, $\overline{v}$ is a pulled-down leaf, and by PULLDOWNLEAVES, $\overline{v} = \kappa(\overline{u}, s)$, where $s \in \hat{\ell}(\overline{u})$. Let u' be the unique node in T' where $\sigma'(u') = \overline{u}$ and $\ell'(u') = s$. If u' is an internal node, TOPULLEDDOWNREFINEMENT adds exactly one edge $(u', \overline{v}')$ where $\overline{\sigma}'(u') = \sigma'(u') = \overline{u}$ and $\overline{\sigma}'(v') = \kappa(\overline{u}, s) = \overline{v}$. On the other hand, if u' is a leaf, TOPULLEDDOWNREFINEMENT modifies the original node of u' to be $\overline{\sigma}'(u') = \kappa(\overline{u}, s) = \overline{v}$. Moreover, since $\kappa(\overline{u}, s) \neq \overline{u}$, by PULLDOWNLEAVES, either $\overline{u}$ is internal in T or $|\hat{\ell}(\overline{u}| > 1$. In both cases, at least one internal node u'' in T' with $\sigma'(u'') = \overline{u}$ must be the parent of u', as $\overline{T}'_{\overline{u}}$ forms a connected subtree. As TOPULLEDDOWNREFINEMENT does not modify internal nodes, $\overline{\sigma}'(u'') = \sigma'(u'') = \overline{u}$.

3. Condition (iii) stating that there are no two nodes in $\overline{T}'_{\overline{v}}$ for any $\overline{v}$ with the same label is satisfied, as (i) if $\overline{v}$ is in T, then $V(\overline{T}'_{\overline{v}}) \subseteq V(T'_{\overline{v}})$ and there are no two nodes in $V(T'_{\overline{v}})$, and (ii) if $\overline{v}$ is not in T, then $T'_{\overline{v}}$ is a single-node tree.

4. Condition (iv) is satisfied because for each leaf $\overline{v}$ in $\overline{T}$ observed in $s \in \hat{\ell}(\overline{v})$, $(\overline{v}, s) \in \mathcal{X}$, and TOPULLEDDOWNREFINEMENT adds a corresponding leaf $\overline{v}'$ in $\overline{T}'$ labeled $\overline{\ell}'(\overline{v}) = s$.

Therefore, $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ is a feasible solution to PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$.

Finally, we must show that $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ is mapped to uniquely by (T', σ', ℓ'). For a contradiction, assume TOPULLEDDOWNREFINEMENT returns $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ for two distinct refinements (T', σ', ℓ') and (T'', σ'', ℓ'') of $(T, \hat{\ell})$. Now, since deleting the nodes location labeled by pulled-down leaves and the corresponding incoming edges added in $\overline{T}'$ returns T' and T'', they are isomorphic. Also, since TOPULLEDDOWNREFINEMENT does not change the location of the nodes simultaneously present in T' or T'', and $\overline{T}'$, it holds that ℓ' and ℓ'' are the same. As the construction also does not change any internal node, it holds that $\sigma'(v') = \sigma''(v')$ for any internal node v' in T' and T''. For a contradiction, assume $\sigma'(v') \neq \sigma''(v')$ for a leaf v'. Without loss of generality, three cases can happen.

- Both $(\sigma'(v'), \ell'(v'))$ and $(\sigma''(v'), \ell''(v'))$ are in $\mathcal{X}(T, \hat{\ell})$. In this case, the construction sets the original node of v' in $\overline{T}'$ to be $\kappa(\sigma'(v'), \ell'(v')) = \kappa(\sigma''(v'), \ell''(v'))$. But this is impossible if $\sigma'(v') \neq \sigma''(v')$.

- $(\sigma'(v'), \ell'(v'))$ is in $\mathcal{X}(T, \hat{\ell})$ but $(\sigma''(v'), \ell''(v'))$ is not. In this case, the construction sets the original node of v' in $\overline{T}'$ to be $\kappa(\sigma'(v'), \ell'(v')) = \sigma''(v')$. But that implies either (i) $\sigma'(v') = \sigma'(v'')$, which violates the premise, or (ii) $\sigma'(v')$ is the parent of $\sigma''(v')$, which is impossible as both are leaves.

- None of $(\sigma'(v'), \ell'(v'))$ and $(\sigma''(v'), \ell''(v'))$ are in $\mathcal{X}$. In this case, the construction sets the original node of v' in $\overline{T}'$ to be $\sigma'(v') = \sigma''(v')$, which violates the premise.

Thus, $\sigma' = \sigma''$, and therefore $(T', \sigma', \ell') = (T'', \sigma'', \ell'')$. □

To conclude the proof of Proposition 1, we present a function FROMPULLEDDOWNREFINEMENT to construct a feasible solution (T', σ', ℓ') to PMH-TR instance $(\mathcal{F}, T, \hat{\ell})$ from $(\overline{T}', \overline{\sigma}', \overline{\ell}')$, $(T, \hat{\ell})$, and $(\overline{T}, \overline{\ell})$, with the same number of migrations and comigrations, i.e. $M(T', \sigma', \ell') = M(\overline{T}', \overline{\sigma}', \overline{\ell}')$ and $C(T', \sigma', \ell') = C(\overline{T}', \overline{\sigma}', \overline{\ell}')$. For each node $\overline{v}' \in V(\overline{T}')$ where (i) the original node $\overline{v} = \overline{\sigma}'(\overline{v}')$ is a pulled-down leaf, i.e. $\overline{v} \in L(\overline{T}) \setminus L(T)$, and (ii) the unique incoming edge $(\overline{u}', \overline{v}')$ to $\overline{v}'$ is a non-migration, i.e. $\overline{\ell}'(\overline{u}') = \overline{\ell}'(\overline{v}')$, the function FROMPULLEDDOWNREFINEMENT constructs the clonal tree T' by deleting node $\overline{v}'$ from $\overline{T}'$ and attaching its children $\overline{w}' \in \delta_{\overline{T}'}(\overline{v}')$ to its parent $\overline{u}' = \pi_{\overline{T}'}(\overline{v}')$. Afterwards, FROMPULLEDDOWNREFINEMENT constructs σ' such that for any node $v' \in V(\overline{T}')$, $\sigma'(v') = \overline{\sigma}'(v')$ if $\overline{\sigma}'(v')$ is in T, or $\sigma'(v') = \overline{\sigma}'(u')$ where u' is the parent of v' in $\overline{T}'$ if the original node $\overline{\sigma}'(v')$ is a pulled-down leaf, i.e. $\overline{\sigma}'(v') \in L(\overline{T}) \setminus L(T)$. Last, FROMPULLEDDOWNREFINEMENT defines $\ell'(v')$ to be $\overline{\ell}'(v')$ for all the nodes v' in T'.

Algorithm S3 Construction of refinement (T', σ', ℓ') of $(T, \hat{\ell})$ from given refinement $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ of $(\overline{T}, \overline{\ell})$

```
procedure FROMPULLEDDOWNREFINEMENT((T, ℓ̂), (T̄', σ̄', ℓ̄'))
    (T̄, ℓ̄) ← PULLDOWNLEAVES(T, ℓ̂)
    (T', ℓ') ← (T̄', ℓ̄')
    for each v̄' ∈ V(T̄') do
        v̄ ← σ̄'(v̄')
        if v̄ ∈ L(T̄) \ L(T) then                                  ▷ v̄ is a pulled-down leaf
            Let ū' be the unique parent of v̄' in T'
            if ℓ̄'(ū') = ℓ̄'(v̄') then
                for each w̄' ∈ δ_T̄'(v̄') do
                    E(T') ← E(T') \ {(v̄', w̄')}
                    E(T') ← E(T') ∪ {(ū', w̄')}
                end for
                V(T') ← V(T') \ {v̄'}
            end if
        end if
    end for
    for each v' ∈ V(T') do
        ℓ'(v') ← ℓ̄'(v')
        if σ̄'(v') ∈ L(T̄) \ L(T) then                              ▷ node σ̄'(v') is a pulled-down leaf
            Let u' be the parent of v' in T̄'
            σ'(v') ← σ̄'(u')
        else                                                      ▷ node σ̄'(v') is not a pulled-down leaf
            σ'(v') ← σ̄'(v')
        end if
    end for
    return (T', σ', ℓ')
end procedure
```

We now prove our second lemma, stating that for $\mathcal{F} \in \{\mathrm{MC}, \mathrm{MC}\}$ the function FROMPULLEDDOWNREFINEMENT with domain PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$ is a surjective function for the range of feasible solutions PMH-TR$^*(\mathcal{F}, T, \hat{\ell})$ with the same number of migrations and comigrations as solutions in PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$.

Lemma 12. *For $\mathcal{F} \in \{\mathrm{MC}, \mathrm{MC}\}$, the function* FROMPULLEDDOWNREFINEMENT *with domain* PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$ *is a surjective function for the range of feasible solutions* PMH-TR$^*(\mathcal{F}, T, \ell)$ *with the same number of migrations and comigrations as solutions in* PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$.

Proof. For any refinement $(\overline{T}', \overline{\sigma}', \overline{\ell}') \in$ PMH-TR$(\mathcal{F}, \overline{T}, \overline{\ell})$, the function call $T', \sigma', \ell') =$ FROMPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (\overline{T}', \overline{\sigma}', \overline{\ell}'))$ attaches children of v' to the parent u' of v' after deleting v, effectively merging the edge (u', v') in tree $\overline{T}'$ to obtain T', T' is a tree. Moreover, clearly ℓ' is valid as it maps

every node in T' to a location from Σ. It remains to show that σ' maps each node in T' to a node in T. Assume for contradiction that node v' in T' is mapped to pulled-down leaf $\overline{u} \in L(\overline{T}) \setminus L(T)$. From FROMPULLEDDOWNREFINEMENT, this is only possible if the parent u' of v' is mapped to $\overline{\sigma}'(u') = \overline{u}$. Again, since the nodes with original labels $\overline{u}$ induce a subtree $\overline{T}'_{\overline{u}}$ in $\overline{T}$ (condition (i) of Definition 2), and $\overline{u}$ is a leaf in $\overline{T}$, v' must be labeled with $\overline{\sigma}'(v') = \overline{u}$ too in $\overline{T}'$. On the other hand, the incoming edge (w', u') in $\overline{T}'$ is a migration, otherwise u' would be deleted. Assume $\overline{u}$ is observed in the unique location s in $\overline{T}$ by PULLDOWNLEAVES. Now, if $\ell'(u') = s$, then deleting migration (u', v') decreases the number of migrations in $\overline{T}'$ by one. Again, if $\ell'(v') = s$, then deleting node u' and attaching v' as a child of w' also decreases the number of migrations in $\overline{T}'$ by one. Since both the cases violate the premise that $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ is optimal, v' in T' cannot have pulled-down leaf $\overline{u} \in L(\overline{T}) \setminus L(T)$ as the original node. Thus, σ' is valid.

Since FROMPULLEDDOWNREFINEMENT only deletes non-migrations from $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ to obtain (T', σ', ℓ'), they have identical sets of migrations and comigrations, i.e. $\mathcal{M}(T', \sigma', \ell') = \mathcal{M}(\overline{T}', \overline{\sigma}', \overline{\ell}')$ and $\mathcal{C}(T', \sigma', \ell') = \mathcal{C}(\overline{T}', \overline{\sigma}', \overline{\ell}')$. By Problem 1, (T', σ', ℓ') is a feasible refinement to PMH-TR instance $(\mathcal{F}, T, \hat{\ell})$ if (T', σ', ℓ') is a refinement of $(T, \hat{\ell})$. To that end, we show that (T', σ', ℓ') meets all the four conditions of refinement from Definition 2.

1. For condition (i), we prove for any original node v in T, the nodes v' in T' where $\sigma'(v')$ induce a connected subtree T'_v in T'. From FROMPULLEDDOWNREFINEMENT, T'_v consists of all the nodes from $\overline{T}'_v$, and the nodes v' with incoming edge (u', v') such that v' has a pulled-down leaf as original node in $\overline{T}'$, i.e. $\overline{\sigma}'(v') \in L(\overline{T}) \setminus L(T)$, (u', v') is a migration, and u' is in $\overline{T}'_v$. Thus, T'_v is a connected subtree in T'.

2. Condition (ii) states that for each edge (u, v) in T, there is a refined edge (u', v') in T' such that $\sigma'(u') = u$ and $\sigma'(v') = v$. Now by PULLDOWNLEAVES, each edge (u, v) of T is present in $\overline{T}$, which ensures exactly one edge $(\overline{u}', \overline{v}')$ in $\overline{T}'$ where $\overline{\sigma}'(\overline{u}') = u$ and $\overline{\sigma}'(\overline{v}') = v$. Since $\overline{\sigma}'(\overline{v}') = v$ is in T, FROMPULLEDDOWNREFINEMENT keeps the edge $(\overline{u}', \overline{v}')$ in T' without changing the corresponding original nodes, i.e. $\sigma'(\overline{u}') = u$ and $\sigma'(\overline{v}') = v$. Thus, condition (ii) is satisfied.

3. Condition (iii) states that for any two nodes v', v'' with the same original node $\sigma'(v') = \sigma'(v'') = v$ have different location labels, i.e. $\ell'(v') \neq \ell'(v'')$. Suppose for contradiction that T'_v has two nodes v', v'' with the same label $\ell'(v') = \ell'(v'') = s$. Since $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ is a refinement of $(\overline{T}, \overline{\ell})$, $\overline{T}'_v$ cannot have two nodes with the same label. Thus, both v' and v'' cannot be in $\overline{T}'_v$. Without loss of generality, assume v' with incoming edge (u', v') is in T'_v but not in $\overline{T}'_v$. Then by FROMPULLEDDOWNREFINEMENT, v' has a pulled-down leaf as original node in $\overline{T}'$, i.e. $\overline{\sigma}'(v') \in L(\overline{T}) \setminus L(T)$, (u', v') is a migration, and u' is in $\overline{T}'_v$. Now deleting the edge (u', v') and adding (v'', v') decreases the number of migrations by one, which contradicts with the premise that $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ is an optimal refinemtnt to PMH-TR instance $(\mathcal{F}, \overline{T}, \overline{\ell})$. So condition (iii) is satisfied.

4. Condition (iv) states that for each node-location pair (v, s) where $s \in \hat{\ell}(v)$, there is a node v' in T'_v such that $\ell'(v') = s$. Assume for any node-location pair (v, s) where $s \in \hat{\ell}(v)$ that $\overline{v} = \kappa(v, s)$ is a leaf in $\overline{T}$ with label $\overline{\ell}(\overline{v}) = s$. Thus there is a node $\overline{v}'$ in $\overline{T}'$ such that $\overline{\sigma}'(\overline{v}') = \overline{v}$ and $\overline{\ell}'(\overline{v}') = s$. By FROMPULLEDDOWNREFINEMENT, $\overline{v}'$ is deleted if the incoming edge $(\overline{u}', \overline{v}')$ to $\overline{v}'$ is a non-migration, which ensures there is a node $\overline{u}'$ labeled by $\overline{\ell}'(\overline{v'}) = \overline{\ell}'(\overline{u'}) = \ell'(\overline{u'}) = s$ in T'_v. On the other hand, if $\overline{v}'$ is not deleted, then $\overline{v}'$ labeled by $\overline{\ell}'(\overline{v'}) = \ell'(\overline{v'}) = s$ is a part of T'_v. Thus, condition (iv) is satisfied.

Therefore, (T', σ', ℓ') is a feasible solution to PMH-TR instance $(\mathcal{F}, T, \hat{\ell})$.

Last, we prove that FROMPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (\overline{T}', \overline{\sigma}', \overline{\ell}'))$ is a surjective function i.e. for any feasible optimal solution (T', σ', ℓ') to PMH-TR$(\mathcal{F}, T, \hat{\ell})$, there is a refinement $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ such that $(T', \sigma', \ell') =$ FROMPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (\overline{T}', \overline{\sigma}', \overline{\ell}'))$. We assume for contradiction that there is no refinement $(\overline{T}', \overline{\sigma}', \overline{\ell}')$ for $(T', \sigma', \ell') \in$ PMH-TR$(\mathcal{F}, T, \hat{\ell})$ such that $(T', \sigma', \ell') =$ FROMPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (\overline{T}', \overline{\sigma}', \overline{\ell}'))$. Let $(\overline{T}', \overline{\sigma}', \overline{\ell}') =$ TOPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (T', \sigma', \ell'))$, and $(T'', \sigma'', \ell'') =$ FROMPULLEDDOWNREFINEMENT$((T, \hat{\ell}), (\overline{T}', \overline{\sigma}', \overline{\ell}'))$. To prove that (T', σ', ℓ') and (T'', σ'', ℓ'') are the same, it suffices to show that for each edge (u', v') in T', (i) (u', v') is in T'', (ii) $\sigma'(u') = \sigma''(u')$ and $\sigma'(v') = \sigma''(v')$, and (iii) $\ell'(u') = \ell''(u')$ and $\ell'(v') = \ell''(v')$.

We begin by showing that for each edge (u', v') in T', (i) (u', v') is in T''. By ToPulledDownRefinement, (u', v') is in $\overline{T}'$. Assume for a contradiction that (u', v') is deleted by FromPulledDownRefinement in T''. This is possible only if $\overline{\sigma}'(v')$ is a pulled-down leaf, and (u', v') is a non-migration. As $\overline{\sigma}'(v')$ is a pulled-down leaf, $|\hat{\ell}(v)| > 1$ or v is an internal node in T, where $v = \sigma'(v')$. But if v is an internal node, v' cannot be labeled with a pulled down leaf. On the other hand, if $|\hat{\ell}(v)| > 1$, then u' must also be in T'_v, and thus (u', v') is a migration, leading to a contradiction. Thus for each edge (u', v') in T', (u', v') is in T''.

Next we show that for any node u' in T', it holds that $\sigma'(u') = \sigma''(u')$. By ToPulledDownRefinement, either $\overline{\sigma}'(u') = \sigma'(u')$, or $\overline{\sigma}'(u') = \kappa(\sigma'(u), \ell'(u))$. In both cases, FromPulledDownRefinement assigns $\sigma''(u') = \sigma'(u')$.

Last, we show that $\ell'(v') = \ell''(v')$ for any node v' in T'. This is true as both ToPulledDownRefinement and FromPulledDownRefinement retain the labels of the clones present in both input and output. Therefore, FromPulledDownRefinement is a surjective function. □

(Main text) Theorem 2. *There exists a bijection between refinements (T', σ', ℓ') and pairs (R, e) of edge-labeled refinement graphs for a given tree T with observed labeling $\hat{\ell}$.*

First, we show that given a pair (R, e) of refinement graph R with edge labeling e, there exists a unique refinement (T', σ', ℓ'). We present a construction of (T', σ', ℓ') from (R, e) given $(T, \hat{\ell})$ in Algorithm S4.

Algorithm S4 Construction of (T', σ', ℓ') from (R, e)

1: **procedure** ConstructTree(T, $\hat{\ell}$, R, e, u)
2: Construct the tree T'_u with nodes $V(T'_u) = \{u'_s \mid s \in R_u\}$ and edges $E(T'_u) = \{(u'_s, u'_t) \mid (s, t) \in E(R_u)\}$
3: $T' \leftarrow T'_u$
4: **for** each $u'_s \in V(T'_u)$ **do**
5: $\ell'(u'_s) \leftarrow s$
6: $\sigma'(u'_s) \leftarrow u$
7: **end for**
8: **for** each $v \in \delta_T(u)$ **do**
9: $(T'[v], \sigma'[v], \ell'[v]) \leftarrow$ ConstructTree($T, \hat{\ell}, R, e, v$)
10: Let (s, t) be the unique edge of R labeled by $e((s, t)) = (u, v)$
11: $V(T') \leftarrow V(T') \cup V(T'[v])$
12: $E(T') \leftarrow E(T') \cup E(T'[v]) \cup \{(u'_s, v'_t)\}$
13: $\ell' \leftarrow \ell' \cup \ell'[v]$
14: $\sigma' \leftarrow \sigma' \cup \sigma'[v]$
15: **end for**
16: **return** (T', σ', ℓ')
17: **end procedure**

Given an edge-labeled refinement graph (R, e), we now prove that ConstructTree($T, \hat{\ell}, R, e, u$) returns a refinement (T', σ', ℓ') for the subtree $T[u]$ of T rooted at u. As such, ConstructTree($T, \hat{\ell}, R, e, r(T)$) will yield a refinement for $(T, \hat{\ell})$. Recall that T'_u is the subtree of T' induced by the nodes

Lemma 13. *For any node $u \in V(T)$, the tuple (T', σ', ℓ') returned by* ConstructTree($T, \hat{\ell}, R, e, u$) *is a refinement for the subtree $T[u]$ of T rooted at node u. Moreover, the refined tree T' is rooted at the node $r(T'_u)$.*

Proof. We present a proof by structural induction on T.

(Base case) For the base case, u is a leaf in T labeled by $\hat{\ell}(u) = s$. We prove the base case by showing that (i) T' is a tree rooted at $r(T'_u)$, and (ii) T' is a refinement of the single node tree $T[u]$. For a leaf u without any children, T' equals T'_u (see Line 2-3 and Line 8 of Algorithm S4), which is clearly isomorphic to R_u (Line 2 of Algorithm S4). Since R_u is a tree by Definition 9, T' is a tree rooted at $r(T'_u)$.

To prove T' to be a refinement of $T[u]$, we show that T' maintains all four conditions of refinement from Definition 2. First, condition (i) states that T'_u induces a tree in T', which is true as $T' = T'_u$. Second, condition (ii) states that for each original edge (u, v) in T, there is exactly one edge (u', v') in T' such that $\sigma'(u') = u$ and $\sigma'(v') = v$. This is trivially satisfied, as $T[u]$ does not have any edge. Third, condition (iii) states that $\ell'(u'_s) \neq \ell'(u'_t)$ for any two

nodes u'_s, u'_t in T'_u, which is satisfied as the construction adds a unique node u'_s with label $\ell'(u'_s) = s$ for each node $s \in V(R_u)$ (Line 2 and Line 5 of Algorithm S4). Fourth, condition (iv) states that there is a node u'_s in T'_u with label $\ell'(u'_s) = s$ for each $s \in \hat{\ell}(u)$. From Definition 9, $\hat{\ell}(u) \subseteq V(R_u)$, which means for each location $s \in \hat{\ell}(u)$, there is a node s in R_u, which adds the node u'_s in T'_u with label $\ell'(u'_s) = s$ (Line 2 and Line 5 of Algorithm S4). Hence, (T', σ', ℓ') is a refinement for the subtree $T[u]$ of T and T' is rooted at $r(T'_u)$.

(Induction step) In this step, we have that u is an internal node of T. The proof proceeds in four steps: showing that (i) T' is a tree rooted at $r(T'_u)$, and (ii) (T', σ') is a refinement of $T[u]$.

First, to prove T' is a tree, notice that T' is initialized in Line 3 by T'_u. Thus we begin by showing that T'_u constructed in Line 2 of CONSTRUCTTREE is a rooted tree. From Line 2, it is clear that T'_u is isomorphic to R_u. Since R_u is a directed tree in R by Definition 9, T'_u is a tree. In Line 4-15, CONSTRUCTTREE iterates over each child v of u in T and adds $T'[v]$ and the edge (u'_s, v'_t) to T', where (s, t) is the unique edge in R labeled by $e((s, t)) = (u, v)$. We show that T' remains a tree rooted at $r(T'_u)$ after each iteration processing node $v \in \delta_T(u)$ (Line 8). By the induction hypothesis, $T'[v]$ is a tree rooted at $r(T'_v)$. The edge (u'_s, v'_t) added in Line 12 of CONSTRUCTTREE connects a node u'_s from T'_u with the root v'_t of $T'[v]$. By Definition 9, if $e((s, t)) = (u, v)$, then $s \in V(R_u)$ (since s has an outgoing edge labeled by (u, v)), which implies that u'_s is a node in T'_u. On the other hand, $e((s, t)) = (u, v)$ implies $t = r(R_v)$, and so v'_t is the root of T'_v since T'_v and R_v are isomorphic, and by extension, the root of $T'[v]$. Thus, T' remains a tree rooted at $r(T'_u)$ after each iteration.

Second, to show that (T', σ', ℓ') is a refinement of $(T, \hat{\ell})$, we prove that the four conditions of refinement from Definition 2 are met.

1. Condition (i) states that T'_v induced by nodes v' where $\sigma'(v') = v$ for any internal node v in $T[u]$ is a tree. Clearly, T'_v generated in Line 2 of CONSTRUCTTREE for any $v \in T[u]$ comprises all nodes v' such that $\sigma'(v') = v$. Given that T'_u is a tree and by induction hypothesis, it follows that T'_v for any internal node v in $T[u]$ is a tree.

2. Condition (ii) states that for each edge (u, v) in T, there is exactly one edge (u'_s, v'_t) such that $\sigma'(u'_s) = u$ and $\sigma'(v'_t) = v$.Note that any edge of T' is either (u'_s, v'_t) where $e((s, t)) = (u, v)$, or in T'_u or $T'[v]$ for any child v of u in T. From CONSTRUCTTREE, clearly $\sigma'(u') = u$ for every node u' in T'_u and there can not be any node w' in $T'[v]$ where $\sigma(w') = u$. Thus there cannot be any edge (u'_s, v'_t) in T'_u or $T'[v]$ such that $\sigma'(u'_s) = u$ and $\sigma'(v'_t) = v$. By Definition 9, there is exactly one edge (s, t) in R labeled by $e((s, t)) = (u, v)$ for each (u, v), which causes exactly one edge (u'_s, v'_t) to be added in T' by CONSTRUCTTREE, where $\sigma'(u'_s) = u$ and $\sigma'(v'_t) = v$.

3. Condition (iii) states that for any two nodes v'_s, v'_t in T'_v for any original node $v \in V(T)$, it holds that $\ell'(u') \neq \ell'(v')$. Now, CONSTRUCTTREE adds a unique node u'_s with label $\ell'(u'_s) = s$ for each node $s \in V(R_u)$ (Line 2 and Line 5), which renders each node of T'_u to have unique label. Thus, given the induction hypothesis, condition (iii) is met for any original node $v \in V(T)$.

4. Condition (iv) states that for each node-location pair (v, s) such that $s \in \hat{\ell}(v)$, there is a refined node v' where the corresponding original node is $\sigma'(v') = v$ and the location label is $\ell'(v') = s$. Now, by condition (iii) of the definition of a refinement graph (Definition 9), $s \in V(R_v)$ if $s \in \hat{\ell}(v)$. Consequently, CONSTRUCTTREE adds a unique node v'_s with label $\ell'(v'_s) = s$ node-location pair (v, s) such that $s \in \hat{\ell}(v)$ (Line 2 and Line 5).

Therefore, (T', σ', ℓ') is a refinement.

□

We next show that given a refinement (T', σ', ℓ') comprised of a refinement (T', σ') and location labeling ℓ', there exists a unique edge-labeled refinement graph (R, e). We present a construction of (R, e) from T',σ', and ℓ' given $(T, \hat{\ell})$ in Algorithm S5. We prove in the following lemma that (R, e) is an edge-labeled refinement graph.

Lemma 14. CONSTRUCTGRAPH*($T', \sigma', \ell', T, \hat{\ell}$) returns an edge-labeled refinement graph (R, e) for $(T, \hat{\ell})$.*

Proof. From CONSTRUCTGRAPH, clearly $V(R) = \Sigma$, and the edge labeling e labels the edges of R either with a clonal tree node from $V(T)$, or a clonal tree edge from $E(T)$. Thus to show that (R, e) is an edge-labeled refinement graph, it suffices to prove that (R, e) adheres to all five conditions from Definition 9.

1. By Definition 2, for each edge (u, v) in T, there exists exactly one edge (u'_s, v'_t) in T' such that $\sigma'(u'_s) = u$ and $\sigma'(v'_t) = v$, and thus CONSTRUCTGRAPH adds exactly one edge (s, t) in R where $\ell'(u'_s) = s$, $\ell'(v'_t) = t$, labeled by $e((s, t)) = (u, v)$, satisfying Condition (i) of Definition 9.

Algorithm S5 Construction of (R, e) from (T', σ', ℓ')

1: **procedure** CONSTRUCTGRAPH($T, \hat{\ell}, T', \sigma', \ell'$)
2: Construct R with nodes $V(R) = \Sigma$ and no edges
3: **for** each node $v \in V(T)$ **do**
4: construct R_v with nodes $V(R_v) = \{\ell'(v'_s) \mid v'_s \in V(T'_v)\}$ and edges $E(R_v) = \{(\ell'(v'_s), \ell'(v'_t)) \mid (v'_s, v'_t) \in E(T'_v)\}$
5: **for** each edge $(s, t) \in E(R_v)$ **do**
6: $e((s, t)) \leftarrow v$
7: **end for**
8: $E(R) \leftarrow E(R) \cup E(R_v)$
9: **end for**
10: **for** each edge $(u, v) \in E(T)$ **do**
11: Let (u'_s, v'_t) be the unique edge of T' where $\sigma'(u'_s) = u$ and $\sigma'(v'_t) = v$
12: $E(R) \leftarrow E(R) \cup \{(\ell'(u'_s), \ell'(v'_t))\}$
13: $e((\ell'(u'_s), \ell'(v'_t))) = (u, v)$
14: **end for**
15: **return** (R, e)
16: **end procedure**

2. Following from the previous point, if v is a leaf in T, then v'_t is a leaf in T' by Definition 2, which implies $\ell'(v'_t) = t = \hat{\ell}(\sigma'(v'_t))$ according to Problem 1. Thus condition (ii) of Definition 9 is fulfilled.

3. R_u is a tree as it is clearly isomorphic to T'_u by Line 4 of CONSTRUCTGRAPH, and the edges (s, t) of R_u are labeled with $e((s, t)) = u$ for any $u \in I(T)$. From CONSTRUCTGRAPH, clearly if node u'_s is in T'_u, then $\ell'(u'_s) = s$ is in R_u. Now since $\sigma'(u'_s) = u$ is internal in T, u'_s cannot be a leaf in T' by Definition 2. As a result, there exists at least one outgoing edge (u'_s, v'_t) in T'. Now if $\sigma'(u'_s) = \sigma'(v'_t) = u$, then edge (u'_s, v'_t) is in T'_u, and so s has an outgoing edge $(s, \ell'(v'_t))$ in isomorphic R_u. On the other hand, if $\sigma'(u'_s) = u \neq \sigma'(v'_t) = v$, then from Definition 2, it follows that (u, v) is an edge in T, and edge (u'_s, v'_t) is the only edge of T' where $\sigma'(u'_s) = u$, $\sigma'(v'_t) = v$. Consequently, CONSTRUCTGRAPH adds edge $(s, \ell'(v'_t))$ labeled by $e((s, \ell'(v'_t))) = (u, v)$ in R. So each node s of R_u has at least one outgoing edge in R labeled either by node $u \in V(T)$ or edge $(u, v) \in E(T)$. Thus condition (iii) of Definition 9 is satisfied.

4. Since (T', σ', ℓ') is a refinement, for any two nodes u'_s, u'_t in T' such that $\sigma'(u'_s) = \sigma'(u'_t) = u$, both u'_s and u'_t have at least two outgoing edges in T'. Now $\sigma'(u'_s) = \sigma'(u'_t)$ implies nodes u'_s, u'_t are in T'_u. In this case, CONSTRUCTGRAPH adds nodes $s = \ell'(u'_s)$ and $t = \ell'(u'_t)$ to R_u. Now if u'_s has an outgoing edge $(u'_s, v'_{t'})$, then CONSTRUCTGRAPH adds edge $(s, \ell'(v'_{t'}))$ and labels it with u if $\sigma'(u'_s) = \sigma'(v'_{t'}) = u$, or with (u, v) if $\sigma'(u'_s) = u \neq \sigma'(v'_{t'}) = v$. Thus for two outgoing edges of u'_s and u'_t, CONSTRUCTGRAPH adds two outgoing edges from s and t labeled either by u or (u, v). Thus condition (iv) of Definition 9 is satisfied.

5. From Definition 2, for any edge (u, v) in T, there exists exactly one edge (u'_s, v'_t) in T' where $\sigma'(u'_s) = u$ and $\sigma(v'_t) = v$. Since T'_v is a subtree of tree T', v'_t must be the root of T'_v. Since T'_v and R_v are isomorphic, $\ell'(v'_t)$ is the root of directed tree R_v and CONSTRUCTGRAPH adds edge $(\ell'(u'_s), \ell'(v'_t))$ labeled by $e((\ell'(u'_s), \ell'(v'_t))) = (u, v)$ in R. Thus, condition (v) of Definition 9 is satisfied.

Therefore, (R, e) is an edge-labeled refinement graph. □

Proposition 15. *Let (R, e) be a refinement graph and (T', σ', ℓ') be a refinement for a given a tree T with leaf labeling $\hat{\ell}$. Then,* CONSTRUCTTREE*($T, \hat{\ell}, R, e, r(T)$) returns tuple (T', σ', ℓ') if and only if* CONSTRUCTGRAPH*($T, \hat{\ell}, T', \sigma', \ell'$) returns (R, e).*

Proof. ($\Rightarrow$) Let (T', σ', ℓ') be the tuple returned by CONSTRUCTTREE($T, \hat{\ell}, R, e, r(T)$). Assume for a contradiction that CONSTRUCTGRAPH($T, \hat{\ell}, T', \sigma', \ell'$) returns a pair (R', e') distinct from (R, e). By Definition 9, each edge (s, t) of R is labeled either by node $v \in V(T)$ or edge $(u, v) \in E(T)$. Now, if any edge (s, t) of R is labeled by $e((s, t)) = u$ for any node $u \in V(T)$, then $(s, t) \in R_u$ by Definition 9. Thus, CONSTRUCTTREE($T, \hat{\ell}, R, e, u$) adds edge (u'_s, u'_t)

to T' and sets $\sigma'(u'_s) = \sigma'(u'_t) = u$, $\ell'(u'_s) = s$, and $\ell'(u'_t) = t$. But then CONSTRUCTGRAPH($T, \hat{\ell}, T', \sigma', \ell'$) adds $(\ell'(u'_s), \ell'(u'_t)) = (s, t)$ labeled by $e'((s, t)) = u$ to R'. On the other hand, if any edge (s, t) of R is labeled by $e((s, t)) = (u, v)$ for any edge $(u, v) \in E(T)$, then CONSTRUCTTREE($T, \hat{\ell}, R, e, u$) adds edge (u'_s, v'_t) to T' and sets $\sigma'(u'_s) = u$, $\sigma'(v'_t) = v$, $\ell'(u'_s) = s$, and $\ell'(v'_t) = t$. But then (u'_s, v'_t) is the unique edge of T' where $\sigma'(u'_s) = u$ and $\sigma'(v'_t) = v$ by Definition 2, and thus CONSTRUCTGRAPH($T, \hat{\ell}, T', \sigma', \ell'$) adds $(\ell'(u'_s), \ell'(v'_t)) = (s, t)$ labeled by $e'((s, t)) = (u, v)$ to R'. Therefore, R and R' are isomorphic and $e = e'$, which leads to a contradiction.

($\Leftarrow$) Let (R, e) be the pair returned by CONSTRUCTGRAPH($T, \hat{\ell}, T', \sigma', \ell'$), and assume for a contradiction that CONSTRUCTTREE($T, \hat{\ell}, R, e, r(T)$) returns (T'', σ'', ℓ'') that is different from (T', σ', ℓ'). Now for any edge (u'_s, u'_t) in T' where $\sigma'(u'_s) = \sigma'(u'_t) = u$ for any $u \in V(T)$ and $\ell'(u'_s) = s, \ell'(u'_t) = t$, it holds that $(u'_s, u'_t) \in E(T'_u)$ by Definition 2. Thus, CONSTRUCTGRAPH($T, \hat{\ell}, T', \sigma', \ell'$) adds edge (s, t) labeled by $e((s, t)) = u$ to R. But then $(s, t) \in E(R_u)$ by Definition 9, and so CONSTRUCTTREE($T, \hat{\ell}, R, e, u$) adds edge (u'_s, u'_t) to T'' and sets $\sigma''(u'_s) = \sigma''(u'_t) = u$, $\ell''(u'_s) = s$, and $\ell''(u'_t) = t$. On the other hand, for any edge (u'_s, v'_t) in T' such that $\sigma'(u'_s) = u \neq \sigma'(v'_t) = v$ and $\ell'(u'_s) = s, \ell'(v'_t) = t$, CONSTRUCTGRAPH($T, \hat{\ell}, T', \sigma', \ell'$) adds edge (s, t) labeled by $e((s, t)) = (u, v)$ to R. But then CONSTRUCTTREE($T, \hat{\ell}, R, e, u$) adds edge (u'_s, v'_t) to T'' and sets $\sigma''(u'_s) = u$, $\sigma''(v'_t) = v$, $\ell''(u'_s) = s$, and $\ell''(v'_t) = t$. Therefore, T' and T'' are isomorphic, and σ'' and ℓ'' are the same as σ' and ℓ' respectively, leading to a contradiction. □

(Main text) Proposition 7. *It holds that $M(T', \sigma', \ell') = |\Sigma| - 1 = |\bigcup_{v \in V(T)} \hat{\ell}(v)| - 1$ if and only if (i) $\ell'(v') \in \Sigma$ for each node v' of T' and (ii) for any two nodes w', w'' of T' where $\ell'(w') = \ell'(w'')$, each node in the undirected path from w' to w'' is labeled with $\ell'(w') = \ell'(w'')$.*

Proof. ($\Rightarrow$) We prove the lemma in three steps — showing that (i) $M(T', \sigma', \ell') = |\Sigma| - 1 = |\bigcup_{v \in V(T)} \hat{\ell}(v)| - 1$ implies there is a bijection between the migrations (u', v') in T' and the locations s from Σ where $s \neq \ell'(r(T'))$, (ii) $\ell'(v') \in \Sigma$ for each node v' of T', and (iii) for any two nodes w', w'' of T' where $\ell'(w') = \ell'(w'')$, each node in the undirected path from w' to w'' is labeled with $\ell'(w') = \ell'(w'')$.

First, we prove that $M(T', \sigma', \ell') = |\Sigma| - 1 = |\{\hat{\ell}(v) \mid v \in L(T)\}| - 1$ implies there is a bijection between the migrations (u', v') in T' and the locations s from Σ where $s \neq \ell'(r(T'))$. To be more specific, we show that there is exactly one unique migration (u', v') for each of $|\Sigma| - 1$ locations $s \in \Sigma \backslash \{\ell'(r(T'))\}$ such that $\ell'(v') = s$, and for each migration (u', v') in T', $\ell'(v')$ is in Σ and not the label of the root $r(T')$. Consider any location $s \in \Sigma$ where s is not the label of the root $r(T')$ and leaf w' in T' is labeled with $\ell'(w') = s$. Then there is at least one migration (u', v') in the path from root $r(T')$ to leaf w' such that $\ell'(v') = s$. Given $M(T', \sigma', \ell') = |\Sigma| - 1$, each of the $|\Sigma| - 1$ locations $s \in \Sigma \backslash \{\ell'(r(T'))\}$ can have exactly one migration (u', v') with $\ell'(v') = s$, and for each migration (u', v') in T', $\ell'(v')$ is in $\Sigma \backslash \{\ell'(r(T'))\}$.

Second, we show that $\ell'(v') \in \Sigma$ for each node v' of T'. Assume for contradiction that $\ell'(v') \notin \Sigma$ for a node v' of T'. Now two cases can happen.

- $v' = r(T')$ **:** Since each location s from $\Sigma \backslash \{\ell'(r(T'))\} = \Sigma \backslash \{\ell'(u')\} = \Sigma$ has at least one migration (u', v') with $\ell'(v') = s$, the number $M(T', \sigma', \ell')$ of migrations is at least $|\Sigma|$, contradicting the premise that $M(T', \sigma', \ell') = |\Sigma| - 1$.

- $v' \neq r(T')$ **:** In this case, $\ell'(r(T')) \in \Sigma$, and there exists at least one migration (u'', v'') in the path from root $r(T')$ to v' such that $\ell'(v'') = \ell'(v')$. Since each of $|\Sigma| - 1$ locations $s \in \Sigma \backslash \{\ell'(r(T'))\}$ has at least one migration (u', v') with $\ell'(v') = s$, the total number $M(T', \sigma', \ell')$ of migrations stands at least $|\Sigma|$ after including the migration (u'', v''), contradicting the premise that $M(T', \sigma', \ell') = |\Sigma| - 1$.

Third, we prove that for any two nodes w', w'' in T' where $\ell'(w') = \ell'(w'')$, each node in the undirected path from w' to w'' is labeled with $\ell'(w') = \ell'(w'')$. Let u' be the most recent common ancestor of any two nodes w', w'' in T' where $\ell'(w') = \ell'(w'')$. Thus, the path from w' to w'' comprises the nodes along the path from u' to w' and from u' to w'', and no edge can be both in the path from u' to w' and from u' to w''. We begin by showing that $\ell'(u') = \ell'(w') = \ell'(w'')$, and then we show the nodes in the path from u' to w' and from u' to w'' are labeled with $\ell'(w') = \ell'(w'')$. Assume for a contradiction that $\ell'(u') \neq \ell'(w') = \ell'(w'')$. Then clearly $u' \neq w'$ and $u' \neq w''$, and therefore, there is at least one migration (u'', v'') in the path from u' to w' and at least one migration (u''', v''') different from (u'', v'') in the path from u' to w'' where $\ell'(v'') = \ell'(w') = \ell'(v''') = \ell'(w'')$. But then there are two migrations (u'', v'') and (u''', v''') where $\ell'(v'') = \ell'(v''')$, contradicting that there is a bijection between the migrations in T' and the locations s from Σ where $s \neq \ell'(r(T'))$. Thus, $\ell'(u') = \ell'(w') = \ell'(w'')$.

($\Leftarrow$) To prove the reverse direction, we assume the conditions in the lemma are true, and prove that $M(T', \sigma', \ell') = |\Sigma| - 1$. From Proposition 6, $M(T', \sigma', \ell') \geq |\Sigma| - 1$. For contradiction, assume $M(T', \sigma', \ell') > |\Sigma| - 1$. Then by the pigeonhole principle, there is at least one location $s \in \Sigma$ such that either (i) there are at least two migrations (u', v') and (u'', v'') such that $\ell'(v') = \ell'(v'') = s \neq \ell'(u')$, or (ii) the root $r(T')$ is labeled with $\ell'(r(T')) = s$ and additionally, there is at least one migrations (u', v') such that $\ell'(v') = s \neq \ell'(u')$. In both cases, node u' is in the path from two nodes labeled by s in tree T', which contradicts with condition (ii) of the premise. Thus, $M(T', \sigma', \ell') = |\Sigma| - 1$. □

(Main text) Proposition 8. *It holds that* $C(T', \sigma', \ell') \leq |E(R)| \leq |E(T)| + |V(T)|(|\Sigma| - 1)$.

Proof. Clearly $C(T', \sigma', \ell') \leq M(T', \sigma', \ell') \leq |E(R)|$, since for each clonal tree edge $(u, v) \in E(T)$, there exists a refinement graph edge (s, t) labeled by $e((s, t)) = (u, v)$ by Definition 9. To show that $|E(R)| \leq |E(T)| + (|V(T)| \times |\Sigma|)$, note that for any clonal tree node $v \in V(T)$, R_v is a tree without any multi-edges by Definition 9. Since $V(G) = \Sigma$, R_v can at most have $|\Sigma|$ nodes, and so there can be at most $|\Sigma| - 1$ edges labeled with v for any $v \in V(T)$. So in R, there are $|E(T)|$ refinement tree edges labeled with a clonal tree edge and at most $|V(T)|(|\Sigma| - 1)$ edges labeled with a clonal tree node. Therefore, $|E(R)| \leq |E(T)| + |V(T)|(|\Sigma| - 1)$. □

(Main text) Proposition 9. *If* M *is the first criterion in* $\mathcal{F}$*, then for any node* $u \in V(T)$ *without polytomies, i.e.* $|\delta_T(u)| < 2$*, it holds that* $|V(T'_u)| = 1$.

Proof. Assume for contradiction that $|V(T'_u)| > 1$ for a node u observed in less than two locations. Therefore by the pigeonhole principle, at least one node u' in T'_u labeled by $\ell'(u') = s$ is is an unobserved clone, i.e $s \notin \hat{\ell}'(u)$. As $|V(T'_u)| > 1$ implies there is at least one edge in connected subtree T'_u, without loss of generality, assume (u', u'') is an edge in T'_u. We construct a new refinement (T'', σ'', ℓ'') by (i) deleting u'', (ii) attaching the children of u'' as the children of u', and (iii) assigning the location of u'' to be the location of u', i.e. $\ell''(u') = \ell'(u'')$. Clearly, $U(T'', \sigma'', \ell'') < U(T', \sigma', \ell')$, which contradicts with the fact that (T'', σ'', ℓ'') is optimal when M is the first criterion in $\mathcal{F}$. □

(Main text) Proposition 10. *Let* (T', σ', ℓ') *be a refinement that minimizes the number of unobserved clones. Then there exists an optimal refinement* (T', σ', ℓ') *such that for any node* $u \in V(T)$ *without polytomies, i.e.* $|\delta_T(u)| < 2$ *and observed in less than two locations, i.e.* $|\hat{\ell}(u)| < 2$*, it holds that* $|V(T'_u)| = 1$.

Proof. Assume (T', σ', ℓ') is an optimal refinement where for a node $u \in V(T)$ without polytomies, i.e. $|\delta_T(u)| < 2$, it holds that $|V(T'_u)| = 1$. Assume (u', u'') is an edge in T_u such that u'' is an edge in T'_u. Note that (u', u'') is a migration by condition (ii) of Definition 2. Now we construct a new refinement (T'', σ'', ℓ'') with $|V(T''_u)| = |V(T'_u)| - 1$ without increasing the number of migrations. We consider three cases.

1. u'' does not have any children. In this case, deleting migration (u', u'') decreases the number of migrations, contradicting the premise.

2. u'' has one child v'. In this case, assigning the label of u'' to v', i.e. $\ell''(v') = \ell'(u'')$ and then collapsing edge (u', u'') reduces the number of migration by one (if (u'', v') is a migration), or keeps the migration number the same (if (u'', v') is not a migration).

3. u'' has two children v' and w'. Since u has at most two children, u' cannot have any outgoing edge outside of T'_u. Thus deleting migration (u', u'') decreases the number of migrations.

Thus we can get a refinement (T'', σ'', ℓ'') that aligns with the premise by repeatedly applying the abovementioned steps. □

B.4 Complexity

In this section, we study the complexity of PMH-TR for different criteria ordering $\mathcal{F}$. To that end, we consider the natural decision and counting versions of the PMH-TR problem, which itself was posed as an optimization problem. Fig. S2c summarizes all the prior and new complexity results. We start with the decision problem.

Problem 2 (DECISION PARSIMONIOUS MIGRATION HISTORY WITH TREE REFINEMENT (DEC-PMH-TR))**.** *Given a clonal tree* T*, set* Σ *of locations, observed labeling* $\hat{\ell} : V(T) \to 2^{|\Sigma|}$*, criteria ordering* $\mathcal{F} = \{f_1, \ldots, f_k\}$ *and cost vector* $(c_1, \ldots, c_k) \in \mathbb{N}$*, does there exist a refinement* (T', σ', ℓ') *of* $(T, \hat{\ell})$ *with cost* $(f_1(T', \sigma', \ell'), \ldots, f_k(T', \sigma', \ell'))$ *at most* $(c_1, \ldots, c_k)$.

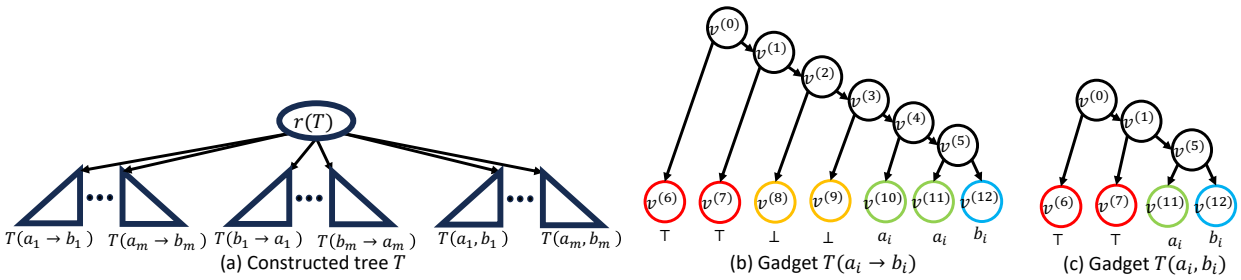


Figure S12: **Reduction from VERTEX COVER.** (a) For each undirected edge (a_i, b_i) in graph $\mathcal{G}$, the constructed clonal tree T contains three gadgets $T(a_i \to b_i)$, $T(b_i \to a_i)$, and $T(a_i, b_i)$. (b) and (c) illustrates the gadgets $T(a_i \to b_i)$ and $T(a_i, b_i)$ with observed labeling $\hat{\ell}$, respectively. Each leaf v is observed in exactly one location, i.e. $|\hat{\ell}(v)| = 1$. Internal nodes v are not observed anywhere, i.e. $|\hat{\ell}(v)| = 0$.

We also consider the counting problem defined as follows.

Problem 3 (COUNTING PARSIMONIOUS MIGRATION HISTORY WITH TREE REFINEMENT (#PMH-TR))**.** *Given a clonal tree T, set Σ of locations, observed labeling $\hat{\ell} : V(T) \to 2^{|\Sigma|}$ and criteria ordering $\mathcal{F}$, count the number of refinements (T', σ', ℓ') of $(T, \hat{\ell})$ that are lexicographically optimal with respect to $\mathcal{F}$.*

B.4.1 Hardness of the PMH-TR Decision Problem

We prove that DEC-PMH-TR is NP-complete when $\mathcal{F} = \{\text{MC}, \text{UMC}, \text{MUC}, \text{MCU}\}$ by reduction from VERTEX COVER, an NP-complete problem [46]. In an undirected graph $\mathcal{G}$ with node set $V(\mathcal{G})$ and edge set $E(\mathcal{G})$, a *vertex cover* K is a subset of nodes $V(\mathcal{G})$ such that for each edge $(a_i, b_i) \in E(\mathcal{G})$, at least one of a_i or b_i is in K. Given $\mathcal{G}$ and an integer k, the VERTEX COVER problem asks if there is vertex cover K^* of size k.

Theorem 16. DEC-PMH-TR *is NP-complete for $\mathcal{F} \in \{\text{MC}, \text{UMC}, \text{MUC}, \text{MCU}\}$.*

We do this by giving a polynomial-time reduction from VERTEX COVER to DEC-PMH-TR. More specifically, given an undirected graph $\mathcal{G}$ with $|V(\mathcal{G})| = n$ nodes and $|E(\mathcal{G})| = m$ edges, we construct a DEC-PMH-TR instance consisting of a clonal tree T with locations $\Sigma = V(\mathcal{G}) \cup \{\bot\}$, and observed labeling $\hat{\ell} : V(T) \to 2^{|\Sigma|}$. The construction ensures that (i) the optimal refinement (T^*, σ^*, ℓ^*) admits $U(T^*, \sigma^*, \ell^*) = 15m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 8m$ migrations and $C(T^*, \sigma^*, \ell^*) = 2m + n + 1 + |K^*|$ comigrations, where $|K^*|$ is the cardinality of a minimum vertex cover K^* for $\mathcal{G}$, and (ii) there is a comigration from $\bot$ to a if node $a \in V(\mathcal{G})$ is in K^*. We satisfy these requirements by using two types of gadgets that form subtrees of T, as illustrated in Fig. S12a.

(i) Gadget $T(a_i \to b_i)$. For each undirected edge (a_i, b_i) in $\mathcal{G}$, we attach two gadgets $T(a_i \to b_i)$ and $T(b_i \to a_i)$ with $\{\top, \bot, a_i, b_i\}$ distinct labels to $r(T)$ in T (Fig. S12b). There are $2m$ such gadgets.

(ii) Gadget $T(a_i, b_i)$. For each undirected edge (a_i, b_i) in $\mathcal{G}$, we attach one gadget $T(a_i, b_i)$ with $\{\top, \bot, a_i, b_i\}$ distinct labels to $r(T)$ in T (Fig. S12c). There are m such gadgets.

Note that in constructed tree T, all the nodes $v^{(k)}$ are observed in either zero or one location, i.e. $0 \leq |\hat{\ell}(u)| < 2$. Therefore, for $\mathcal{F} = \text{UMC}$, it holds that $|V(T^*_{v^{(k)}})| = 1$ for each node $v^{(k)}$ by Proposition 9. On the other hand, only the root $r(T)$ has more than two children. Thus, by Proposition 10, there exists an optimal refinement (T', σ', ℓ') such that for each non-root node $v^{(k)}$, $|V(T'_{v^{(k)}})| = 1$ when $\mathcal{F} \in \{\text{MC}, \text{MCU}, \text{MUC}\}$, and in this case, we assume (T^*, σ^*, ℓ^*) to be such a refinement. Let $v'^{(k)}$ be the unique node mapping to the original non-root node $\sigma^*(v'^{(k)}) = v^{(k)}$. By applying Lemma 7 to the gadgets of T', we prove that (T^*, σ^*, ℓ^*) has $8m$ migrations and determine the location label of all the nodes in gadgets $T(a_i \to b_i)$ and the nodes $v'^{(0)}$ and $v'^{(1)}$ of gadgets $T(a_i, b_i)$.

Lemma 17. *For any optimal refinement (T^*, σ^*, ℓ^*) for criteria ordering $\mathcal{F} \in \{\text{MC}, \text{UMC}, \text{MUC}, \text{MCU}\}$, it holds that*

(i) $U(T^*, \sigma^*, \ell^*) = 15m$,

(ii) $M(T^*, \sigma^*, \ell^*) = 8m$,

(iii) for each node v' in a gadget, there is a leaf w' in the same gadget such as $\ell^*(v') = \ell'(w')$,

(iv) $\ell^*(v'^{(0)}) = \ell^*(v'^{(1)}) = \top$ for every gadget,

(v) $\ell^*(v'^{(2)}) = \ell^*(v'^{(3)}) = \bot$ for gadgets $T(a_i \to b_i)$,

(vi) $T'_{r(T)}$ contains exactly one node $r(T')$ and $\ell^*(r(T')) = \top$.

Proof. To prove (i)-(ii), first we prove that $U(T^*, \sigma^*, \ell^*) \geq 15m$ and $M(T^*, \sigma^*, \ell^*) \geq 8m$. Note that $U(T^*, \sigma^*, \ell^*) \geq 15m$ directly follows from applying Lemma 5 to (T^*, σ^*, ℓ^*). To prove $M(T^*, \sigma^*, \ell^*) \geq 8m$, we apply Lemma 6 to each gadget and find the minimum number of migrations in T^*.

- Each gadget $T^*(a_i \to b_i)$ has 4 distinct leaf labels $\{\top, \bot, a_i, b_i\}$, and so has a minimum of $4-1 = 3$ migrations. Thus, $2m$ gadgets $T^*(a_i \to b_i)$ contribute at least $3 \cdot 2m = 6m$ migrations in T^*.
- Each gadget $T^*(a_i, b_i)$ has 3 distinct leaf labels $\{\top, a_i, b_i\}$, and so has a minimum of $3 - 1 = 2$ migrations. Thus, m gadgets $T^*(a_i, b_i)$ contribute at least $2m$ migrations in T^*.

Totalling, $M(T^*, \sigma^*, \ell^*) \geq 8m$.

Now to prove that the optimal refinement (T^*, σ^*, ℓ^*) for $\mathcal{F} \in \{\mathrm{MC}, \mathrm{UMC}, \mathrm{MUC}, \mathrm{MCU}\}$ has exactly $15m$ unobserved clones and $8m$ migrations, it suffices to show that there exists at least one refinement (T', σ', ℓ') that simultaneously reaches the lower limit for unobserved clones $U(T^*, \sigma^*, \ell^*) = 15m$ and the number of migrations $M(T', \sigma', \ell') = 8m$. We construct such a (T', σ', ℓ') as follows.

- set $\ell'(v'^{(0)}) = \ell'(v'^{(1)}) = \top$ for every gadget,
- set $\ell'(v'^{(2)}) = \ell'(v'^{(3)}) = \bot$ and $\ell'(v'^{(4)}) = \ell'(v'^{(5)}) = a_i$ for gadgets $T(a_i \to b_i)$.
- set $\ell'(v'^{(5)}) = a_i$ for gadgets $T(a_i, b_i)$.

Clearly (T', σ', ℓ') has $15m$ unobserved clones and $8m$ migrations. Therefore, $U(T^*, \sigma^*, \ell^*) = 15m$ and $M(T^*, \sigma^*, \ell^*) = 8m$.

To prove (iii)-(v), notice that (T^*, σ^*, ℓ^*) achieves the minimum number $8m$ of migrations if each gadget has the minimum number of migrations. As such, by Lemma 7, for each node v' in a gadget, there is a leaf w' in the same gadget such as $\ell^*(v') = \ell'(w')$. Moreover, if two leaves w' and w'' in a gadget in T' have the same location label, i.e. $\ell^*(w') = \ell^*(w'')$, then each node v' in the (undirected) path from w' to w'' has the same label $\ell^*(v') = \ell^*(w')$. Now in every gadget, $\ell^*(v'^{(6)}) = \ell^*(v'^{(7)}) = \top$, and the path between them contains $v'^{(0)}$ and $v'^{(1)}$. Thus, $\ell'(v'^{(0)}) = \ell'(v'^{(1)}) = \top$. Similarly, $\ell'(v'^{(2)}) = \ell'(v'^{(3)}) = \bot$ and $\ell'(v'^{(4)}) = \ell'(v'^{(5)}) = a_i$ for gadgets $T(a_i \to b_i)$, since $v'^{(2)}$ and $v'^{(3)}$ are in the path between $v'^{(8)}$ and $v'^{(9)}$ where $\ell'(v'^{(8)}) = \ell'(v'^{(9)}) = \bot$, and $v'^{(4)}$ and $v'^{(5)}$ are in the path between $v'^{(10)}$ and $v'^{(11)}$ where $\ell'(v'^{(10)}) = \ell'(v'^{(11)}) = a_i$.

We finish proving that the lemma is true for $\mathcal{F} \in \{\mathrm{MC}, \mathrm{MCU}, \mathrm{MUC}, \mathrm{UMC}\}$ by showing that $T^*_{r(T)}$ contains exactly one node $r(T^*)$ and $\ell^*(r(T^*)) = \top$. As $|\hat{\ell}(r(T))| = 0$, by Proposition 9, $|V(T^*_{r(T)})| = 1$ for $\mathcal{F} = \mathrm{UMC}$. For $\mathcal{F} \in \{\mathrm{MC}, \mathrm{MCU}, \mathrm{MUC}\}$, notice that $M(T^*, \sigma^*, \ell^*) = 8m$ implies every migration is inside a gadget, every gadget has the minimum number of migrations, and the root $v'^{(0)}$ of each gadget is labeled by $\top$. Now if there is a node v' such that $\sigma^*(v') = r(T)$ and $\ell^*(v') \neq \top$, then there is at least one migration in the path from v' to the root $v'^{(0)}$ of any gadget in T^*. But that puts migration outside the gadgets, which increases the number of migrations, thereby reaching a contradiction. So there is exactly one node $r(T')$ such that $\sigma^*(r(T')) = r(T)$, and $\ell^*(r(T')) = \top$. □

Based on Lemma 17, we can get a list of comigrations that are present in (T^*, σ^*, ℓ^*).

Corollary 18. *For any optimal refinement (T^*, σ^*, ℓ^*) for criteria ordering $\mathcal{F} \in \{\mathrm{MC}, \mathrm{UMC}, \mathrm{MUC}, \mathrm{MCU}\}$, there are comigrations from*

- *$\top$ to $\bot$,*
- *$\bot$ to a_i for each $a_i \in V(\mathcal{G})$,*
- *a_i to b_i and b_i to a_i for each $(a_i, b_i) \in E(\mathcal{G})$.*

Finally, we prove that the size of the minimum vertex cover for graph $\mathcal{G}$ can be determined from the number $C(T^*, \sigma^*, \ell^*)$ of comigrations.

Proposition 19. *There exists a minimum vertex cover K^* of size $|K^*| = k$ if and only if there exists an optimal refinement (T^*, σ^*, ℓ^*) for criteria ordering $\mathcal{F} \in \{\text{MC}, \text{UMC}, \text{MUC}, \text{MCU}\}$ with $U(T^*, \sigma^*, \ell^*) = 15m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 8m$ migrations and $C(T^*, \sigma^*, \ell^*) = n + 2m + 1 + k$ comigrations.*

Proof. ($\Rightarrow$) Assume there is a refinement (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 15m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 8m$ migrations and $C(T^*, \sigma^*, \ell^*) = n + 2m + 1 + k$ comigrations. Let us consider the gadget $T^*(a_i, b_i)$. $T^*(a_i, b_i)$ has distinct leaf labels $\{\top, a_i, b_i\}$, and node $v'^{(1)}$ in $T^*(a_i, b_i)$ is labeled with $\ell^*(v'^{(1)}) = \top$. Now by condition (iii) of Lemma 17, $v'^{(5)}$ is labeled with one of $\top$, a_i, or b_i. If $v'^{(5)}$ is labeled with $\top$, then there exists comigrations from $\top$ to both a_i and b_i, which is not listed in Corollary 18. Again, if $v'^{(5)}$ is labeled with a_i or b_i, then there exists comigrations from $\top$ to either a_i or b_i not listed in Corollary 18. Let K be the set of nodes from $\mathcal{G}$ such that a_i is in K if there is a migration from $\top$ to a_i in the gadget $T(a_i, b_j)$.

Now for each edge (a_i, b_i) in $\mathcal{G}$, $T^*(a_i, b_i)$ has at least one comigration from $\top$ to either a_i or b_i or both, which implies that at least one of a_i or b_i is in K. Thus K is a vertex cover for $\mathcal{G}$. Again, Corollary 18 lists $n + 2m + 1$ comigrations present in (T^*, σ^*, ℓ^*). Thus, (T^*, σ^*, ℓ^*) has $(n + 2m + 1 + k) - (n + 2m + 1) = k$ comigrations not listed in Corollary 18. Since all the comigrations in (T^*, σ^*, ℓ^*) are either listed in Corollary 18 or from $\top$ to an element from K, $|K| = k$.

($\Leftarrow$) Assume that K^* is a minimum vertex cover for the undirected graph $\mathcal{G}$. We show that there exists an optimal refinement with $U(T^*, \sigma^*, \ell^*) = 15m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 8m$ migrations and $C(T^*, \sigma^*, \ell^*) = n + 2m + 1 + k$ comigrations. We construct one such (T^*, σ^*, ℓ^*) as follows.

1. Let T^* have the same topology as T.
2. Set the mapping function σ^* such that σ^* induces a bijection between the nodes of T and T^*.
3. Following Lemma 17, set $\ell^*(r(T')) = \ell^*(v'^{(0)}) = \ell^*(v'^{(1)}) = \top$ for every gadget, and set $\ell^*(v'^{(2)}) = \ell^*(v'^{(3)}) = \bot$ and $\ell^*(v'^{(4)}) = \ell^*(v'^{(5)}) = a_i$ for gadgets $T^*(a_i \to b_i)$.
4. For every edge (a_i, b_i) from $\mathcal{G}$ where a_i is in vertex cover K^*, set $\ell^*(v'^{(5)}) = a_i$ in gadget $T^*(a_i, b_i)$.

Clearly, (T^*, σ^*, ℓ^*) is a refinement with $U(T^*, \sigma^*, \ell^*) = 15m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 8m$ migrations and $C(T^*, \sigma^*, \ell^*) = n + 2m + 1 + k$ comigrations. □

Finally, we prove our second theorem about NP-completeness for criteria orderings $\{\text{CM}, \text{UCM}, \text{CMU}\}$.

Theorem 20. DEC-PMH-TR *is NP-complete for $\mathcal{F} \in \{\text{CM}, \text{CMU}, \text{UCM}\}$.*

Proof. We show the hardness of PMH-TR separately for each criteria ordering $\{\text{CM}, \text{CMU}, \text{UCM}\}$.

1. $F = \text{CM}$. PMH-TR is proven to be NP-complete for $F = \text{CM}$ in El-Kebir et al. [41].
2. $F = \text{CMU}$. Notice that PMH-TR solution space for $F = \text{CMU}$ is a subset of the solution space for $F = \text{CM}$. Therefore, PMH-TR for $F = \text{CM}$ can be trivially reduced to PMH-TR for $F = \text{CMU}$, as if there is a refinement with at most c comigrations and m migrations, then there is a refinement with at most c comigrations, m migrations, and $|\Sigma| \dot{|} V(T)|$ unobserved clones.
3. $F = \text{UCM}$. This problem can be reduced from PMH for $F = \text{CM}$, proven to be NP-complete in El-Kebir et al. [41]. In PMH for $F = \text{CM}$, given a clonal tree T with leaf labeling $\hat{\ell} : L(T) \to \Sigma$, and two integers c and m, one asks if there is a location labeling ℓ' that minimizes the number of comigrations first, and then migrations. In other words, given $(T, \hat{\ell})$ and two integers c and m, PMH seeks a refinement (T', σ', ℓ') such that $|V(T'_v)| = 1$ for each node v, $C(T', \sigma', \ell') = c$, and $M(T', \sigma', \ell') = m$. As $\hat{\ell}$ labels clones with less than two locations, the constraint $|V(T'_v)| = 1$ can be enforced by requiring minimizing the number of unobserved clones first, as suggested by Proposition 5. Therefore, the solution space for PMH instance $(T, \hat{\ell})$ for criteria ordering $F = \text{CM}$ and the solution space for PMH-TR instance $(T, \hat{\ell})$ for criteria ordering $F = \text{UCM}$ are identical. Thus, PMH-TR is NP-complete for $F = \text{UCM}$.

□

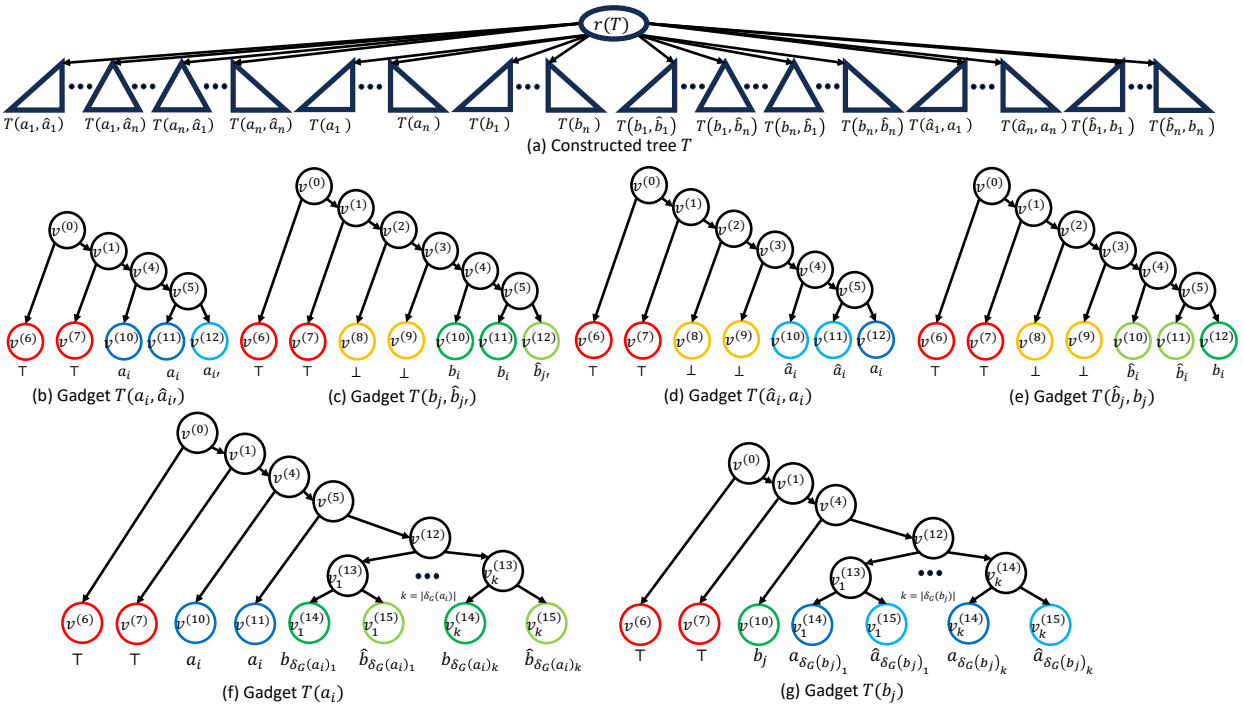


Figure S13: **Construction of the clonal tree T given a bipartite graph $\mathcal{G}$ with $V(\mathcal{G}) = A \cup B$ nodes.** (a) The constructed tree T containing six types of gadget is shown. (b-g) illustrates six types of gadgets.

B.4.2 Hardness of the #PMH-TR Counting Problems

We prove that PMH-TR is #P-complete when $\mathcal{F} \in \{\text{UMC}, \text{MUC}, \text{MCU}\}$ by providing a strong parsimony reduction from the problem of counting the number of perfect matchings in a bipartite graph, which itself is #P-complete [47]. A graph $\mathcal{G}$ with nodes $V(\mathcal{G})$ and edges $E(\mathcal{G})$ is bipartite if its nodes $V(\mathcal{G})$ can be partitioned into two disjoint sets A and B such that each edge $(a_i, b_j) \in E(\mathcal{G})$ connects a node a_i from A to a node b_j from B. A perfect matching M in a bipartite graph $\mathcal{G}$ is a subset of edges $E(\mathcal{G})$ such that each node a_i or b_j in $V(\mathcal{G}) = A \cup B$ is incident to exactly one edge (a_i, b_j) in M. Though determining if there is a perfect matching in a given bipartite graph is polynomial-time solvable, counting the number of perfect matchings is #P-complete [47].

Theorem 21. #PMH-TR *is #P-complete for* $\mathcal{F} \in \{\text{UMC}, \text{MUC}, \text{MCU}\}$.

Clearly, #PMH-TR is in #P as the underlying decision problem (where one additionally specifies the set of comigrations as a partition of migrations) is in NP, i.e. one can verify correctness of the solution in polynomial time. To prove #P-hardness, we show a polynomial-time parsimonious reduction from perfect matching in a bipartite graph to PMH-TR for criteria ordering $\mathcal{F} \in \{\text{UMC}, \text{MUC}, \text{MCU}\}$. To that end, consider a bipartite graph $\mathcal{G}$ with $V(\mathcal{G}) = A \cup B$ nodes and $E(\mathcal{G})$ edges, where $|A| = |B| = n$, $|E(\mathcal{G})| = m$, $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$. Let the function $\delta_{\mathcal{G}} : V(\mathcal{G}) \to 2^{|V(\mathcal{G})|}$ indicate node adjacency in $\mathcal{G}$ such that $\delta_{\mathcal{G}}(a_i) = b_j$ and $\delta_{\mathcal{G}}(b_j) = a_i$ if (a_i, b_j) is in $\mathcal{G}$. In addition, we declare new symbols $\widehat{A} = \{\hat{a}_1, \ldots, \hat{a}_n\}$ and $\widehat{B} = \{\hat{b}_1, \ldots, \hat{b}_n\}$. Given the bipartite graph $\mathcal{G}$, we construct a clonal tree T with leaf labeling $\hat{\ell} : L(T) \to \Sigma$ that labels the leaves of T with locations $\Sigma = \{\top, \bot\} \cup A \cup B \cup \widehat{A} \cup \widehat{B}$ such that the comigrations induced by each optimal refinement (T^*, σ^*, ℓ^*) encodes a perfect matching K of $\mathcal{G}$. Specifically, we want to ensure that (i) for each optimal refinement (T^*, σ^*, ℓ^*) there are exactly $10n^2 + 21n + 2m$ unobserved clones, $5n^2 + 8n + 4m$ migrations and at least $2n^2 + 6n + 1$ comigrations, and (ii) there is a perfect matching K if (T^*, σ^*, ℓ^*) has exactly $2n^2 + 6n + 1$ comigrations and edge (a_i, b_j) is in K if there is a comigration from a_i to b_j. We accomplish this by using six different types of gadget that for the subtrees of T, as shown in Fig. S13.

(i) Gadget $T(a_i, \hat{a}_{i'})$. For each $1 \leq i, i' \leq n$, we attach one gadget $T(a_i, \hat{a}_{i'})$ with $\{\top, a_i, \hat{a}_{i'}\}$ distinct labels to the root $r(T)$ (Fig. S13b). There are n^2 such gadgets.

(ii) Gadget $T(b_j, \hat{b}_{j'})$. For each $1 \leq j, j' \leq n$, we attach one gadget $T(b_j, \hat{b}_{j'})$ with $\{\top, \bot, b_j, \hat{b}_{j'}\}$ distinct labels to the root $r(T)$ (Fig. S13c). There are n^2 such gadgets.

(iii) Gadget $T(\hat{a}_i, a_i)$. For each $1 \leq i \leq n$, we attach one gadget $T(\hat{a}_i, a_i)$ with $\{\top, \bot, a_i, \hat{a}_i\}$ distinct labels to the root $r(T)$ (Fig. S13d). There are n such gadgets.

(iv) Gadget $T(\hat{b}_j, b_j)$. For each $1 \leq j \leq n$, we attach one gadget $T(\hat{b}_j, b_j)$ with $\{\top, \bot, b_j, \hat{b}_j\}$ distinct labels to the root $r(T)$ (Fig. S13e). There are n such gadgets.

(v) Gadget $T(a_i)$. For each $1 \leq i \leq n$, we attach one gadget $T(a_i)$ with $\{\top, a_i\} \cup \delta_{\mathcal{G}}(a_i) \cup \{\hat{b}_j \mid b_j \in \delta_{\mathcal{G}}(a_i)\}$ distinct labels to the root $r(T)$ (Fig. S13f). There are n such gadgets.

(vi) Gadget $T(b_j)$. For each $1 \leq j \leq n$, we attach one gadget $T(b_j)$ with $\{\top, b_j\} \cup \delta_{\mathcal{G}}(b_j) \cup \{\hat{a}_i \mid a_i \in \delta_{\mathcal{G}}(b_j)\}$ distinct labels to the root $r(T)$ (Fig. S13g). There are n such gadgets.

Note that all the nodes $v^{(k)}$ of T are observed in either zero or one location, i.e. $0 \leq |\hat{\ell}(v^{(k)})| < 2$. Therefore, for $\mathcal{F} = \mathrm{UMC}$, it holds that $|V(T^*_{v^{(k)}})| = 1$ for each node $v^{(k)}$ (Proposition 9). Moreover, in constructed tree T, only the root $r(T)$ and $v^{(12)}$ in gadgets $T(a_i)$ and $T(b_j)$ have more than two children. Thus, by Proposition 10, there exists an optimal refinement (T', σ', ℓ') such that for each node $v^{(k)}$ where $v^{(k)} \neq r(T)$ and $v^{(k)} \neq v^{(12)}$, $|V(T'_{v^{(k)}})| = 1$ when $\mathcal{F} \in \{\mathrm{MCU}, \mathrm{MUC}\}$, and in this case, we assume (T^*, σ^*, ℓ^*) to be such a refinement. Additionally, for each node $v^{(k)}$ where $v^{(k)} \neq r(T)$ and $v^{(k)} \neq v^{(12)}$, we let $v'^{(k)}$ be the unique node mapping to the original node $\sigma^*(v'^{(k)}) = v^{(k)}$. By applying Lemma 5 and Lemma 7 to the gadgets of T', we prove that (T^*, σ^*, ℓ^*) has $10n^2 + 21n + 2m$ unobserved clones and $5n^2 + 8n + 4m$ migrations, and determine the location label of all the nodes in gadgets $T(a_i, \hat{a}_{i'})$, $T(b_j, \hat{b}_{j'})$, $T(\hat{a}_i, a_i)$, $T(\hat{b}_j, b_j)$, the nodes $v'^{(0)}$, $v'^{(1)}$, $v'^{(2)}$, $v'^{(3)}$ in gadgets $T(a_i)$, and the nodes $v'^{(0)}$, $v'^{(1)}$ in gadgets $T(b_j)$.

Lemma 22. *For any optimal refinement (T^*, σ^*, ℓ^*) for criteria ordering $\mathcal{F} \in \{\mathrm{UMC}, \mathrm{MUC}, \mathrm{MCU}\}$, it holds that*

(i) $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$,

(ii) $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$,

(iii) for each node u' in a gadget, if there exists a leaf w' in the same gadget where $\ell^*(u') = \ell^*(w')$, then the nodes v' in the path from u' to w' are labeled with $\ell^*(v') = \ell^*(u')$,

(iv) $\ell^*(v'^{(0)}) = \ell^*(v'^{(1)}) = \top$ *and* $\ell^*(v'^{(2)}) = \ell^*(v'^{(3)}) = \bot$ *for every gadget,*

(v) both $\ell^*(v'^{(4)})$ and $\ell^*(v'^{(5)})$ equals a_i, b_i, $\hat{a}_i$, $\hat{b}_i$, and a_i for gadgets $T(a_i, \hat{a}_j)$, $T(b_i, \hat{b}_j)$, $T(\hat{a}_i, a_i)$, $T(\hat{b}_i, b_i)$, and $T(a_i)$, respectively,

(vi) $T'_{r(T)}$ contains exactly one node $r(T')$ and $\ell^*(r(T')) = \top$.

Proof. First we prove that $U(T^*, \sigma^*, \ell^*) \geq 10n^2 + 21n + 2m$ and $M(T^*, \sigma^*, \ell^*) \geq 5n^2 + 8n + 4m$. To that end, we apply Lemma 5 and Lemma 7 to each gadget and find the minimum number of migrations in T^*.

- Each gadget $T^*(a_i, \hat{a}_{i'})$ has 4 clones not observed anywhere, and so has a minimum of 4 unobserved clones. Moreover, $T^*(a_i, \hat{a}_{i'})$ has 3 distinct leaf labels $\{\top, a_i, \hat{a}_{i'}\}$, and so has a minimum of $3 - 1 = 2$ migrations. n^2 gadgets $T^*(a_i, \hat{a}_{i'})$ contribute $4n^2$ unobserved clones and $2n^2$ migrations in T^*.

- Each gadget $T^*(b_j, \hat{b}_{j'})$ has 6 clones not observed anywhere, and so has a minimum of 6 unobserved clones. Moreover, $T^*(b_j, \hat{b}_{j'})$ has 4 distinct leaf labels $\{\top, \bot, b_j, \hat{b}_{j'}\}$, and so has a minimum of $4 - 1 = 3$ migrations. n^2 gadgets $T^*(b_j, \hat{b}_{j'})$ contribute $6n^2$ unobserved clones and $3n^2$ migrations in T^*.

- Each gadget $T^*(\hat{a}_i, a_i)$ has 6 clones not observed anywhere, and so has a minimum of 6 unobserved clones. Moreover, $T^*(\hat{a}_i, a_i)$ has 4 distinct leaf labels $\{\top, \bot, a_i, \hat{a}_i\}$, and so has a minimum of $4 - 1 = 3$ migrations. n gadgets $T^*(\hat{a}_i, a_i)$ contribute $6n$ unobserved clones and $3n$ migrations in T^*.

- Each gadget $T^*(\hat{b}_j, b_j)$ has 6 clones not observed anywhere, and so has a minimum of 6 unobserved clones. Moreover, $T^*(\hat{b}_j, b_j)$ has 4 distinct leaf labels $\{\top, \bot, b_j, \hat{b}_j\}$, and so has a minimum of $4 - 1 = 3$ migrations. n gadgets $T^*(\hat{b}_j, b_j)$ contribute $6n$ unobserved clones and $3n$ migrations in T^*.

- Each gadget $T^*(a_i)$ has $5 + |\delta_G(a_i)|$ clones not observed anywhere, and so has a minimum of $5 + |\delta_G(a_i)|$ unobserved clones. Moreover, $T^*(a_i)$ has $2+2|\delta_G(a_i)|$ distinct leaf labels $\{\top, a_i\} \cup \delta_G(a_i) \cup \{\hat{b}_j \mid b_j \in \delta_G(a_i)\}$, and so has a minimum of $2 + 2|\delta_G(a_i)| - 1 = 2|\delta_G(a_i)| + 1$ migrations. Thus, gadgets $T^*(a_i)$ contribute $\sum_{a_i \in A}(5 + |\delta_G(a_i)|) = 5n + m$ unobserved clones and $\sum_{a_i \in A}(2|\delta_G(a_i)| + 1) = n + 2m$ migrations.
- Each gadget $T^*(b_j)$ has $4 + |\delta_G(b_j)|$ clones not observed anywhere, and so has a minimum of $4 + |\delta_G(b_j)|$ unobserved clones. Moreover, $T^*(b_j)$ has $2+2|\delta_G(b_j)|$ distinct leaf labels $\{\top, b_j\} \cup \delta_G(b_j) \cup \{\hat{a}_i \mid a_i \in \delta_G(b_j)\}$, and so has a minimum of $2 + 2|\delta_G(b_j)| - 1 = 2|\delta_G(b_j)| + 1$ migrations. Thus, gadgets $T^*(b_j)$ contribute $\sum_{b_j \in B}(4 + |\delta_G(b_j)|) = 4n + m$ unobserved clones and $\sum_{b_j \in B}(2|\delta_G(b_j)| + 1) = n + 2m$ migrations.

Totalling, $U(T^*, \sigma^*, \ell^*) \geq 10n^2 + 21n + 2m$ and $M(T^*, \sigma^*, \ell^*) \geq 5n^2 + 8n + 4m$.

Now to prove that the optimal refinement (T^*, σ^*, ℓ^*) has exactly $10n^2 + 21n + 2m$ unobserved clones and $5n^2 + 8n + 4m$ migrations, it suffices to show that there exists at least one PMH-TR refinement (T', σ', ℓ') such that $U(T', \sigma', \ell') = 10n^2 + 21n + 2m$ and $M(T', \sigma', \ell') = 5n^2 + 8n + 4m$. We construct such a (T', σ', ℓ').

- set $\ell'(v'^{(0)}) = \ell'(v'^{(1)}) = \top$ and $\ell'(v'^{(2)}) = \ell'(v'^{(3)}) = \perp$ for every gadget,
- set both $\ell'(v'^{(4)})$ and $\ell'(v'^{(5)})$ to be a_i, b_j, $\hat{a}_i$, $\hat{b}_j$, and a_i for gadgets $T(a_i, \hat{a}_{i'})$, $T(b_j, \hat{b}_{j'})$, $T(\hat{a}_i, a_i)$, $T(\hat{b}_j, b_j)$, and $T(a_i)$, respectively,
- let $T'_{r(T)}$ contain exactly one node $r(T')$ and set $\ell'(r(T')) = \top$,
- in gadgets $T'(a_i)$, set $\ell'(v^{(12)})$ and $\ell'(v_k^{(13)})$ for any $1 \leq k \leq |\delta_G(a_i)|$ to be a_i, and in gadgets $T'(b_j)$, set $\ell'(v^{(4)})$, $\ell'(v_k^{(12)})$ for any $1 \leq k \leq |\delta_G(b_j)|$, and $\ell'(v^{(13)})$ to be b_j.

Clearly T' has $10n^2 + 21n + 2m$ unobserved clones and $5n^2 + 8n + 4m$ migrations. Therefore, $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ and $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$.

To prove (iii)-(v), notice that (T^*, σ^*, ℓ^*) achieves the minimum number $5n^2 + 8n + 4m$ of migrations if each gadget has the minimum number of migrations. As such, by Lemma 7, for each node v' in a gadget, there is a leaf w' in the same gadget such as $\ell^*(v') = \ell'(w')$. Moreover, if two leaves w' and w'' in a gadget in T' have the same location label, i.e. $\ell^*(w') = \ell^*(w'')$, then each node v' in the (undirected) path from w' to w'' has the same label $\ell^*(v') = \ell^*(w')$. Now in every gadget, $\ell^*(v'^{(6)} = \ell^*(v'^{(7)} = \top$, and the path between them contains $v'^{(0)}$ and $v'^{(1)}$. Thus, $\ell'(v'^{(0)}) = \ell'(v'^{(1)}) = \top$. Similarly we can prove that $\ell'(v'^{(2)}) = \ell'(v'^{(3)}) = \perp$ for every gadget, and both $\ell^*(v'^{(4)})$ and $\ell^*(v'^{(5)})$ equals a_i, b_j, $\hat{a}_i$, $\hat{b}_j$, and a_i for gadgets $T(a_i, \hat{a}_{i'})$, $T(b_j, \hat{b}_{j'})$, $T(\hat{a}_i, a_i)$, $T(\hat{b}_j, b_j)$, and $T(a_i)$, respectively.

Last, we prove that $T'_{r(T)}$ contains exactly one node $r(T')$ and $\ell^*(r(T')) = \top$. As $|\hat{\ell}(r(T))| = 0$, by Proposition 9, $|V(T^*_{r(T)})| = 1$ for $\mathcal{F} = \text{UMC}$. For $\mathcal{F} \in \{\text{MC}, \text{MCU}, \text{MUC}\}$, notice that $|M(T^*, \ell^*)| = 5n^2 + 8n + 4m$ implies every migration is inside a gadget, every gadget has the minimum number of migrations, and the root $v'^{(0)}$ of each gadget is labeled by $\top$. Now if there is a node v' such that $\sigma^*(v') = r(T)$ and $\ell^*(v') \neq \top$, then there is at least one migration in the path from v' to the root $v'^{(0)}$ of any gadget in T^*. But that puts migration outside the gadgets, which increases the number of migrations, thereby reaching a contradiction. So there is exactly one node $r(T')$ such that $\sigma^*(r(T')) = r(T)$, and $\ell^*(r(T')) = \top$. □

Based on Lemma 22, we can get a list of comigrations that are present in (T^*, σ^*, ℓ^*).

Corollary 23. *For any optimal refinement (T^*, σ^*, ℓ^*) for criteria ordering $\mathcal{F} \in \{\text{UMC}, \text{MUC}, \text{MCU}\}$, there are comigrations from*

- $\top$ *to* $\perp$,
- $\top$ *to* a_i *for any* $1 \leq i \leq n$,
- $\perp$ *to* $\hat{a}_i$ *for any* $1 \leq i \leq n$,
- a_i *to* $\hat{a}_{i'}$ *for any* $1 \leq i, i' \leq n$,
- $\hat{a}_i$ *to* a_i *for any* $1 \leq i \leq n$,

- $\perp$ *to* b_j *for any* $1 \leq j \leq n$,
- $\perp$ *to* $\hat{b}_j$ *for any* $1 \leq j \leq n$,
- b_j *to* $\hat{b}_{j'}$ *for any* $1 \leq j, j' \leq n$,
- $\hat{b}_j$ *to* b_j *for any* $1 \leq j \leq n$.

Last, we prove that there is a bijection between perfect matchings K and optimal refinements (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations.

Proposition 24. *There is a bijection between the set of perfect matchings K of G and the set of optimal refinements (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations.*

Proof. We prove the lemma in four steps: (i) we show that $C(T^*, \sigma^*, \ell^*) \geq 2n^2 + 6n + 1$ for any optimal refinement (T^*, σ^*, ℓ^*), which implies that any refinement (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations is optimal, (ii) we prove that for any optimal refinement (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations, there is a unique perfect matching K such that (T^*, σ^*, ℓ^*) has a comigration from a_i to b_j for each edge (a_i, b_j) in K, (iii) given a perfect matching K, we construct an optimal refinement (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations that admits a comigration from a_i to b_j for each edge (a_i, b_j) in K, and (iv) we prove that the constructed refinement (T^*, σ^*, ℓ^*) as the only refinement admitting a comigration from a_i to b_j for each edge (a_i, b_j) in K.

First, we prove $C(T^*, \sigma^*, \ell^*) \geq 2n^2 + 6n + 1$. Note that there are at least $2n^2 + 5n + 1$ comigrations in (T^*, σ^*, ℓ^*) listed in Corollary 23. Let us consider the gadget $T^*(a_i)$. $T^*(a_i)$ has distinct leaf labels $\{\top, a_i\} \cup \delta_G(a_i) \cup \{\hat{b}_j \mid b_j \in \delta_G(a_i)\}$, and node $v'^{(5)}$ in $T^*(a_i)$ is labeled with $\ell^*(v'^{(5)}) = a_i$. Since $v'^{(5)}$ has descendant leaves $v'^{(14)}$ and $v'^{(15)}$ such that $\ell^*(v'^{(14)}) \neq a_i$ and $\ell^*(v'^{(15)}) \neq a_i$, there is at least one migration (v', v'') in the path from $v'^{(5)}$ to $v'^{(14)}$ or $v'^{(15)}$ such that $\ell^*(v') = a_i$. Now by condition (iii) of Lemma 22, $\ell^*(v'') \in \{\top\} \cup \delta_G(a_i) \cup \{\hat{b}_j \mid b_j \in \delta_G(a_i)\}$. Thus, each of n gadgets $T(a_i)$ has a unique comigration from a_i to one of $\top$, b_j or $\hat{b}_j$ where $b_j \in \delta_G(a_i)$, which is not listed in Corollary 23. Let the set of these n distinct comigrations be $\mathcal{A}$. Adding these comigrations from $\mathcal{A}$ with the comigrations listed in Corollary 23, we get $C(T^*, \sigma^*, \ell^*) \geq 2n^2 + 6n + 1$.

Second, we prove that for any optimal refinement (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations, there is a unique perfect matching K such that (T^*, σ^*, ℓ^*) has a comigration from a_i to b_j for each edge (a_i, b_j) in K. To that end, let us consider the comigrations of the gadgets $T^*(b_j)$ with distinct leaf labels $\{\top, b_j\} \cup \delta_G(b_j) \cup \{\hat{a}_i \mid a_i \in \delta_G(b_j)\}$. Since node $v'^{(3)}$ is labeled with $\ell^*(v'^{(3)}) = \top$ and leaf $v'^{(10)}$ is labeled with $\ell^*(v'^{(10)}) = b_j$, there is at least one migration (v', v'') in the path from $v'^{(3)}$ to $v'^{(10)}$ such that $\ell^*(v'') = b_j$. By condition (iii) of Lemma 22, $\ell^*(v') \in \{\top\} \cup \delta_G(b_j) \cup \{\hat{a}_i \mid a_i \in \delta_G(b_j)\}$. Thus, each of n gadgets $T^*(b_j)$ has a comigration from one of $\top$, a_i or $\hat{a}_i$ to b_j, where $a_i \in \delta_G(b_j)$. Let the set of these n distinct comigrations be $\mathcal{B}$.

Now (T^*, σ^*, ℓ^*) has $(2n^2 + 6n + 1) - (2n^2 + 5n + 1) = n$ comigrations not listed in Corollary 23, which includes comigrations from $\mathcal{A}$ and $\mathcal{B}$. Since each of $\mathcal{A}$ and $\mathcal{B}$ has n distinct comigrations, $\mathcal{A}$ must be equal to $\mathcal{B}$. But $\mathcal{A} = \mathcal{B}$ is possible only if both includes comigrations only from a_i to b_j such that $(a_i, b_j) \in E(\mathcal{G})$. Let K be a subset of $E(\mathcal{G})$ such that $(a_i, b_j) \in K$ if there is a comigrations from a_i to b_j. Now by construction, $\mathcal{A} = \mathcal{B}$ has exactly one comigration from each node a_i in A, and exactly one comigration to each node b_j in B, which implies each node a_i or b_j from $A \cup B$ is incident to exactly one edge in K. Therefore, K is a perfect matching for bipartite graph $\mathcal{G}$.

Third, given a perfect matching K, we construct an optimal refinement (T^*, σ^*, ℓ^*) with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations that admits a comigration from a_i to b_j for each edge (a_i, b_j) in K. The steps are as follows.

1. Set the refinement (T^*, σ^*, ℓ^*) such that T and T^* have the same topology and σ^* induces a bijection between the nodes of T and T^*. Let $v'^{(12)}$ be the unique node such that $\sigma^*(v'^{(12)}) = v^{(12)}$.

2. Following Lemma 22, set $\ell^*(r(T')) = \ell^*(v'^{(0)}) = \ell^*(v'^{(1)}) = \top$ and $\ell^*(v'^{(2)}) = \ell^*(v'^{(3)}) = \bot$ for every gadget, and set both $\ell^*(v'^{(4)})$ and $\ell^*(v'^{(5)})$ to be a_i, b_i, $\hat{a}_i$, $\hat{b}_i$, and a_i for gadgets $T(a_i, \hat{a}_j)$, $T(b_i, \hat{b}_j)$, $T(\hat{a}_i, a_i)$, $T(\hat{b}_i, b_i)$, and $T(a_i)$, respectively.

3. If edge (a_i, b_j) is in K, set $\ell^*(v'^{(12)}) = b_j$ in gadget $T(a_i)$. Moreover, for any $1 \le k \le |\delta_G(a_i)|$, set $\ell^*(v'^{(13)_k}) = b_j$ if $v'^{(14)_k}$ is labeled with b_j, or set $\ell^*(v_k'^{(13)}) = \hat{b}_{j'}$ if $v'^{(14)_k}$ is labeled with $b_{j'} \neq b_j$. Similarly, if edge (a_i, b_j) is in K, set $\ell^*(v'^{(4)}) = \ell^*(v'^{(12)}) = a_i$ in gadget $T(b_j)$. Moreover, for any $1 \le k \le |\delta_G(b_j)|$, set $\ell^*(v'^{(13)_k}) = a_i$ if $v'^{(14)_k}$ is labeled with a_i, or set $\ell^*(v_k'^{(13)}) = \hat{a}_{i'}$ if $v'^{(14)_k}$ is labeled with $a_{i'} \neq a_i$.

Clearly, (T^*, σ^*, ℓ^*) is an optimal refinement with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations and admits comigration from a_i to b_j for each edge (a_i, b_j) in M.

Finally, we prove bijection by showing the constructed refinement (T^*, σ^*, ℓ^*) as the only refinement admitting a comigration from a_i to b_j for each edge (a_i, b_j) in K. Assume for contradiction that the optimal refinement (T', σ', ℓ') with $U(T^*, \sigma^*, \ell^*) = 10n^2 + 21n + 2m$ unobserved clones, $M(T^*, \sigma^*, \ell^*) = 5n^2 + 8n + 4m$ migrations, and $C(T^*, \sigma^*, \ell^*) = 2n^2 + 6n + 1$ comigrations, is different from (T^*, σ^*, ℓ^*) and admits a comigration from a_i to b_j for each edge (a_i, b_j) in K. Since both (T^*, σ^*, ℓ^*) and (T', σ', ℓ') are optimal, they both follow the conditions mentioned in Lemma 22. As such, $T^*_{r(T)} = T'_{r(T)}$, $T^*(a_i, \hat{a}_{i'}) = T'(a_i, \hat{a}_{i'})$, $T^*(b_j, \hat{b}_j) = T'(b_j, \hat{b}_j)$, $T^*(\hat{a}_i, a_i) = T'(\hat{a}_i, a_i)$, $T^*(\hat{b}_j, b_j) = T'(\hat{b}_j, b_j)$ for any $1 \le i, i', j, j' \le n$. Moreover, $\ell^*(v'^{(0)}) = \ell^*(v'^{(1)}) = \ell'(v'^{(0)}) = \ell'(v'^{(1)}) = \top$ and $\ell^*(v'^{(2)}) = \ell^*(v'^{(3)}) = \ell'(v'^{(2)}) = \ell'(v'^{(3)}) = \bot$ in the refined gadgets of $T(a_i)$, and $\ell^*(v'^{(0)}) = \ell^*(v'^{(1)}) = \ell'(v'^{(0)}) = \ell'(v'^{(1)}) = \top$ in the refined gadget of $T(b_j)$ for any $1 \le i, j \le n$. Thus, to show that (T^*, σ^*, ℓ^*) and (T', σ', ℓ') cannot be different and thus lead to contradiction, the following statements must be true.

(i) For any node a_i in A where $(a_i, b_j) \in K$ and for any $1 \le k \le |\delta_G(a_i)|$, it holds that $\ell'(v_k'^{(13)}) = b_j$ if $v_k'^{(12)}$ is labeled with b_j, or $\ell^*(v_k'^{(13)}) = \hat{b}_{j'}$ if $v_k'^{(12)}$ is labeled with $b_{j'} \neq b_j$ in gadget $T'(a_i)$. Similarly, for any node b_j in B where $(a_i, b_j) \in K$ and for any $1 \le k \le |\delta_G(b_j)|$, $v_k'^{(13)}$ is labeled with a_i if $\ell'(v_k'^{(12)}) = a_i$, or $v_k'^{(13)}$ is labeled with $\hat{a}_{i'}$ if $\ell'(v_k'^{(12)}) = a_{i'} \neq a_i$ in gadget $T'(b_j)$.

(ii) For any node a_i in A, $T'_{v'^{(12)}}$ has only one node $v'^{(12)}$ and is labeled with a_i in gadget $T'(a_i)$.

(iii) For any node b_j in B where $(a_i, b_j) \in M$, $\ell^*(v'^{(4)}) = \ell^*(v'^{(12)}) = a_i$ in gadget $T'(b_j)$.

To prove (i), we consider the node $v'^{(13)}$ from the gadget $T'(a_i)$ without loss of generality. Note that $v'^{(13)}$ has two leaves with labels b_j and $\hat{b}_j$, where b_j is the k-th element in $\delta_G(a_i)$.

So $v'^{(13)}$ is labeled either with b_j or $\hat{b}_j$. Now two cases can happen.

- $(a_i, b_j) \in K$: In this case, there must be a comigration (u', v') from a_i to b_j in the path from $v'^{(5)}$ to $v'^{(14)}$ in $T'(a_i)$ such that the nodes in the path from v' to $v'^{(14)}$ are labeled with b_j. Since labeling $v'^{(13)}$ with a_i admits new comigration from a_i to $\hat{b}_j$ and thus makes (T', σ', ℓ') suboptimal, $\ell'(v'^{(13)}) = b_j$.

- $(a_i, b_j) \notin K$: In this case, there can only be comigration to b_j from $\hat{b}_j$ in $T'(a_i)$. Thus if $\ell'(v'^{(13)}) \notin \{b_j, \hat{b}_j\}$, then (T', σ', ℓ') has a new comigration to b_j and is suboptimal. On the other hand, if $v'^{(13)}$ is labeled with b_j, then there is a a migration (u', v') such that $\ell'(u') = \hat{b}_j$ and $\ell'(v') = b_j$ in the path from $v'^{(5)}$ to $v'^{(14)}$ in $T'(a_i)$. But that implies the two nodes v' and $v'^{(14)}$ labeled with b_j are disconnected, which means (T', σ', ℓ') is suboptimal. Thus, $\ell'(v'^{(13)}) = \hat{b}_j$.

Similarly, we can show that for any node b_j in B where $(a_i, b_j) \in K$ and for any $1 \le k \le |\delta_G(b_j)|$, $v_k'^{(13)}$ is labeled with a_i if $\ell'(v_k'^{(14)}) = a_i$, or $v_k'^{(13)}$ is labeled with $\hat{a}_{i'}$ if $\ell'(v_k'^{(14)}) = a_{i'} \neq a_i$ in gadget $T'(b_j)$. Therefore, (i) is true.

Next, we prove that for any node a_i in A, $T'_{v'^{(12)}}$ has only one node $v'^{(12)}$ and is labeled with b_j in gadget $T'(a_i)$ where $(a_i, b_j) \in K$. To that end, we first prove $v'^{(12)} = r(T'_{v'^{(12)}})$ is labeled with b_j. Note that $v'^{(5)}$, the parent of the root of $T'_{v'^{(12)}}$, is labeled with a_i. Since there is a comigration from a_i to b_j, $v'^{(12)} = r(T'_{v'^{(12)}})$ is labeled either with a_i or with b_j. Assume $\ell'(v'^{(12)}) = a_i$. Now two cases can happen.

- $T'_{v'^{(12)}}$ has only one node $v'^{(12)}$: In this case, there are comigrations from a_i to $\hat{b}_j$, where $(a_i, b_j) \notin M$, which are not in the $2n^2 + 6n + 1$ comigrations mentioned above. Thus (T', σ', ℓ') increases the number of comigrations and is not optimal.

- $T'_{v'^{(12)}}$ has at least two nodes: Since $|\hat{\ell}(T'_{v'^{(12)}})| = 0$, $T'_{v'^{(12)}}$ having two nodes increases the number of unobserved clones, making (T', σ', ℓ') suboptimal.

Therefore, $\ell'(v'^{(12)}) \neq a_i$, and so $\ell'(v'^{(12)}) = b_j$.

Next, we prove that for any node b_j in B where $(a_i, b_j) \in K$, $\ell^*(v'^{(4)}) = \ell^*(v'^{(12)}) = a_i$ in gadget $T'(b_j)$. Note that there is a comigration from a_i to b_j in $T'(b_j)$ if $(a_i, b_j) \in M$. Since $\ell'(v'^{(1)}) = \top$, $\ell'(v'^{(10)}) = b_j$, and $v'^{(4)})$ is the only node between $v'^{(1)})$ and $v'^{(10)})$, $v'^{(4)}$ must be labeled with a_i to admit the comigration from a_i to b_j. Let $v_k'^{(14)}$ be the unique leaf in $T'(b_j)$ labeled with a_i. Since $r(T'_{v^{(12)}}) = v'^{(12)}$ is in the path from $v'^{(4)})$ to $v_k'^{(14)}$ where both are labeled with a_i, $v'^{(12)}$ must be labeled by a_i to keep the number of migrations minimum.

Therefore, $(T^*, \sigma^*, \ell^*) = (T', \sigma', \ell')$. □

C MACH2 Integer Linear Program

We start by modeling (R, e) with binary variables $\mathbf{x} \in \{0,1\}^{|\Sigma|\times|\Sigma|\times(|V(T)|+|E(T))|}$. Specifically, $x_{s,t,(u,v)} = 1$ if a refinement graph edge (s,t) is labeled by clonal tree edge $e((s,t)) = (u,v)$ and $x_{s,t,(u,v)} = 0$ otherwise. Similarly, $x_{s,t,v} = 1$ if a refinement graph edge (s,t) is labeled by clonal tree node $e((s,t)) = v$ and $x_{s,t,v} = 0$ otherwise. To ensure $\mathbf{x}$ encodes an edge-labeled refinement graph (R, e) for a pair $(T, \hat{\ell})$ of clonal tree T and observed labeling $\hat{\ell}$, we impose constraints that model all four conditions of a refinement graph (Definition 9). Condition (i) of Definition 9 states that the subgraph R_u consisting of nodes $V(R_u) = \{s \mid e((s,t)) = u \text{ or } e((s,t)) = (u,v) \text{ or } e((s',s)) = u \text{ or } e((s',s)) = (u',u)\}$ and edges $E(R_u) = \{(s,t) \mid e((s,t)) = u\}$ induces a directed tree without multi-edges in R. As this condition does not directly translate to ILP constraints, we resort to the following lemma.

Lemma 25. *For any node $u \in V(T)$, the subgraph R_u induced in R is a directed tree without multi-edges if there is a root node s such that (i) s does not have any incoming edge (s', s) in R_u, i.e. $e((s',s)) \neq u$ for any $(s',s) \in R$; (ii) for any location $t \in V(R_u)$ where $t \neq s$, there is exactly one incoming edge (t', t) in R_u (i.e. labeled by $e((t',t)) = u$); and (iii) there is a path from s to any location $t \in V(R_u)$ where $t \neq s$ in R_u.*

Proof. ($\Rightarrow$) From the premise, R_u is a tree rooted at s. Then every node t of R_u that is not the root i.e. $t \neq s$, has exactly one incoming edge (s', t) in R_u labeled by $e((s',t)) = u$, since R_u only contains edges labeled by u. For the same reason, the root s does not have any incoming edge labeled with u. Last, since R_u is a tree, there is a directed path from the root s to other non-root nodes t in R_u.

($\Leftarrow$) Assume that the conditions in the lemma are true. To prove that R_u is a tree rooted at s, we show that there is exactly one directed path from root s to each other node t in R_u. Assume for a contradiction that there exists one node t in R_u such that there are at least two directed paths from s to t in R_u. But then there exists at least one common node s' in both the paths from s to t, which implies s' has two incoming edges in R_u, which contradicts with the premise. Thus, there is exactly one directed path from s to each other node t in R_u, and so R_u is a tree rooted at s. □

Condition (ii) of Lemma 25 characterizes the non-root nodes t of R_u to be the ones with exactly one incoming edge in R_u. More specifically, t is a non-root node in R_u if $\sum_{t' \in \Sigma} x_{t',t,v}$ is 1. Moreover, condition (iv) of Definition 9 states that the root of R_u is s if u is not the root of T and there is an incoming edge (s', s) in R labeled by $e((s',s)) = (\pi_T(u), u)$. In other words, s is the root of R_u if $\sum_{s' \in \Sigma} x_{s',s,(\pi(v),v)}$ is 1. Note that Definition 9 does not characterize the root $R_{r(T)}$ of R_u if u is the root of T. To that end, we introduce a new binary variable $\mathbf{w} \in \{0,1\}^{|\Sigma|}$ that models the root of $R_{r(T)}$ such that $w_s = 1$ if the root of $R_{r(T)}$ is $s \in \Sigma$, and $w_s = 0$ otherwise. The following constraint ensures that the root of $R_{r(T)}$ is unique.

$$\sum_{s \in \Sigma} w_s = 1.$$

We enforce conditions (i) and (ii) from Lemma 25 requiring the root node s in R_u to have no incoming edges and each non-root node t to have exactly one incoming edge in R_u as follows.

$$\sum_{s' \in \Sigma} x_{s',s,v} \leq 1 - \sum_{s' \in \Sigma} x_{s',s,(\pi(v),v)}, \qquad \forall s \in \Sigma, v \in V(T) \setminus \{r(T)\},$$

$$\sum_{s' \in \Sigma} x_{s',s,r(T)} \leq 1 - w_s, \qquad \forall s \in \Sigma.$$

To enforce condition (iii) of Lemma 25, we introduce two new binary variables $\mathbf{y} = \{0,1\}^{|\Sigma|\times|\Sigma|\times|V(T)|}$ and $\mathbf{z} = \{0,1\}^{|\Sigma|\times|\Sigma|\times|V(T)|}$ to model the relations between any two nodes in R_u for any original node $u \in V(T)$. More specifically, we require $y_{s,t,u}$ to be 1 if there is a directed path from location s to t in R_u, and $y_{s,t,u} = 0$ otherwise. The following constraints enforce $y_{s,t,u} = 1$ under this condition but do not explicitly enforce $y_{s,t,u} = 0$ otherwise.

$$\begin{aligned} y_{s,s,v} &= 0, && \forall s \in \Sigma, v \in V(T), \\ y_{s,t,v} &\geq x_{s,t,v}, && \forall s,t \in \Sigma, v \in V(T), \\ y_{s,t,v} &\geq y_{s,s',v} + x_{s',t,v} - 1, && \forall s,s',t \in \Sigma, v \in V(T). \end{aligned}$$

We require $z_{s,t,u}$ to be 1 if s and t are in R_u, but there is no directed path from s to t or from t to s in R_u, and $z_{s,t,u} = 0$ otherwise. The following constraints set $z_{s,t,u} = 1$ under these conditions but do not explicitly enforce $z_{s,t,u} = 0$ otherwise.

$$\begin{aligned} z_{s,s,v} &= 0, && \forall s \in \Sigma, v \in V(T), \\ z_{s,t,v} &\geq x_{s',s,v} + x_{s',t,v} - 1, && \forall s,s',t \in \Sigma, v \in V(T), \\ z_{s,t,v} &\geq z_{s,s',v} + x_{s',t,v} - 1, && \forall s,s',t \in \Sigma, v \in V(T). \end{aligned}$$

The following constraints enforce the entries in $\mathbf{y}$ and $\mathbf{z}$ to be 0 when applicable by requiring that two locations s and t are in R_u if and only if either there is a path from s to t, or from t to s, or there are no paths between s and t in R_u.

$$\begin{aligned} y_{s,t,v} + y_{t,s,v} + z_{s,t,v} &\leq \sum_{s' \in \Sigma} \left(x_{s',s,v} + x_{s',s,(\pi(v),v)} \right), && \forall s,t \in \Sigma, v \in V(T) \setminus r(T), \\ y_{s,t,r(T)} + y_{t,s,r(T)} + z_{s,t,r(T)} &\leq w_s + \sum_{s' \in \Sigma} x_{s',s,v}, && \forall s,t \in \Sigma, \\ y_{s,t,v} + y_{t,s,v} + z_{s,t,v} &\leq \sum_{t' \in \Sigma} \left(x_{t',t,v} + x_{t',t,(\pi(v),v)} \right), && \forall s,t \in \Sigma, v \in V(T) \setminus r(T), \\ y_{s,t,r(T)} + y_{t,s,r(T)} + z_{s,t,r(T)} &\leq w_t + \sum_{t' \in \Sigma} x_{t',t,v}, && \forall s,t \in \Sigma, \\ y_{s,t,v} + y_{t,s,v} + z_{s,t,v} &\geq \sum_{s' \in \Sigma} \left(x_{s',s,v} + x_{s',s,(\pi(v),v)} \right) \\ &\quad + \sum_{t' \in \Sigma} \left(x_{t',t,v} + x_{t',t,(\pi(v),v)} \right) - 1, && \forall s,t \in \Sigma, v \in V(T) \setminus r(T), \\ y_{s,t,r(T)} + y_{t,s,r(T)} + z_{s,t,r(T)} &\geq w_s + \sum_{s' \in \Sigma} x_{s',s,v} \\ &\quad + w_t + \sum_{t' \in \Sigma} x_{t',t,v} - 1, && \forall s,t \in \Sigma. \end{aligned}$$

We enforce condition (iii) of Lemma 25 with the following constraints.

$$\begin{aligned} y_{s,t,v} &\geq \sum_{s' \in \Sigma} x_{s',s,(\pi(v),v)} + \sum_{t' \in \Sigma} x_{t',t,v} - 1, && \forall s,t \in \Sigma, v \in (T) \setminus \{r(T)\}, \\ y_{s,t,r(T)} &\geq w_s + \sum_{t' \in \Sigma} x_{t',t,r(T)} - 1, && \forall s,t \in \Sigma. \end{aligned}$$

To conclude enforcing condition (i) of Definition 9, we require s' to be in R_u for each edge (s',t') labeled by $e((s',t')) = (u,v)$ where v is a child of u.

$$\begin{aligned} \sum_{s' \in \Sigma} \left(x_{s',s,u} + x_{s',s,(\pi(u),u)} \right) &\geq \sum_{t \in \Sigma} x_{s,t,(u,v)}, && \forall s \in \Sigma, u \in V(T) \setminus \{r(T)\}, v \in \delta_T(u), \\ w_s + \sum_{s' \in \Sigma} x_{s',s,r(T)} &\geq \sum_{t \in \Sigma} x_{s,t,(u,v)}, && \forall s \in \Sigma, v \in \delta_T(r(T)). \end{aligned}$$

The following constraint ensures condition (ii) of Definition 9 by allowing exactly one edge (s,t) of R to be labeled by $e((s,t)) = (u,v)$ for each edge (u,v) in T.

$$\sum_{s,t\in\Sigma} x_{s,t,(u,v)} = 1, \qquad \forall (u,v) \in E(T).$$

Last, we add the following constraints to model condition (iii) of a refinement graph R with edge labeling e from Definition 9, ensuring that for each pair (v,s) of node $v \in L(T)$ and location s where $s \in \hat{\ell}(v)$, $s \in V(R_v)$.

$$\sum_{t\in\Sigma} \left(x_{t,s,(\pi_T(v),v)} + x_{t,s,v}\right) = 1, \qquad \forall v \in L(T)\backslash r(T), s \in \hat{\ell}(v),$$

$$w_s + \sum_{t\in\Sigma} x_{t,s,r(T)} = 1, \qquad \forall s \in \hat{\ell}(r(T)).$$

Note that condition (iv) of Definition 9 is already enforced.

For the objective function, we count the number of unobserved clones as follows.

$$U(T',\sigma',\ell') = \sum_{v\in V(T)} \sum_{s\in\Sigma\backslash\hat{\ell}(v)} \sum_{t\in\Sigma} \left(x_{t,s,(\pi_T(v),v)} + x_{t,s,v}\right)$$

Next, we first compute the number of migrations. From Corollary 4, it follows that $\mathcal{M}(T',\sigma',\ell') = \{(s,t) \in E(R) \mid s \neq t\}$. Thus, in ILP, we compute the number of migrations as follows.

$$M(T',\sigma',\ell') = \sum_{s,t\in\Sigma: s\neq t} \sum_{k\in V(T)\cup E(T)} x_{s,t,k}.$$

We do not compute the number of comigrations in the ILP, rather we use PARSIMONIOUS CONSISTENT COMIGRATIONS (PCC) algorithm [34] to compute the number of comigrations and eliminate solutions from the solution space that exceed the minimum. Based on the discussion above, we propose the following minimization functions for ILP.

$$\begin{aligned}
U(T',\sigma',\ell') + \frac{M(T',\sigma',\ell')}{|E(T) + V(T)(|\Sigma| - 1))}, &\qquad \text{if prefix}(\mathcal{F}) = \text{UM},\\
M(T',\sigma',\ell') + \frac{U(T',\sigma',ell')}{|V(T)| \times |\Sigma|)}, &\qquad \text{if prefix}(\mathcal{F}) = \text{MU},\\
U(T',\sigma',\ell') &\qquad \text{if prefix}(\mathcal{F}) = \text{U},\\
M(T',\ell') &\qquad \text{if prefix}(\mathcal{F}) = \text{M}.
\end{aligned}$$

The factors $|E(T) + V(T)(|\Sigma| - 1))$ and $\frac{1}{|V(T)|\times|\Sigma|)}$ ensure the ILP performs lexicographical minimization following the specified criteria ordering $\mathcal{F}$. We state the full ILP in Section C.

C.1 Complete ILP

We present the ILP for $\mathcal{F} = \text{UM}$ here.

$$
\begin{aligned}
\min \quad & \sum_{s,t\in\Sigma: s\neq t} \sum_{k\in V(T)\cup E(T)} x_{s,t,k} + \frac{\sum_{s,t\in\Sigma} y_{s,t}}{|E(T)| + |I(T)|(|\Sigma|-1)}. \\
\text{s.t.} \quad & \sum_{s\in\Sigma} w_s = 1, \\
& \sum_{s'\in\Sigma} x_{s',s,v} \leq 1 - \sum_{s'\in\Sigma} x_{s',s,(\pi(v),v)}, && \forall s\in\Sigma, v\in V(T)\setminus\{r(T)\}, \\
& \sum_{s'\in\Sigma} x_{s',s,r(T)} \leq 1 - w_s, && \forall s\in\Sigma, \\
& y_{s,s,v} = 0, && \forall s\in\Sigma, v\in V(T), \\
& y_{s,t,v} \geq x_{s,t,v}, && \forall s,t\in\Sigma, v\in V(T), \\
& y_{s,t,v} \geq y_{s,s',v} + x_{s',t,v} - 1, && \forall s,s',t\in\Sigma, v\in V(T), \\
& z_{s,s,v} = 0, && \forall s\in\Sigma, v\in V(T), \\
& z_{s,t,v} \geq x_{s',s,v} + x_{s',t,v} - 1, && \forall s,s',t\in\Sigma, v\in V(T), \\
& z_{s,t,v} \geq z_{s,s',v} + x_{s',t,v} - 1, && \forall s,s',t\in\Sigma, v\in V(T), \\
& y_{s,t,v} + y_{t,s,v} + z_{s,t,v} \leq \sum_{s'\in\Sigma} \left(x_{s',s,v} + x_{s',s,(\pi(v),v)}\right), && \forall s,t\in\Sigma, v\in V(T)\setminus r(T), \\
& y_{s,t,r(T)} + y_{t,s,r(T)} + z_{s,t,r(T)} \leq w_s + \sum_{s'\in\Sigma} x_{s',s,v}, && \forall s,t\in\Sigma, \\
& y_{s,t,v} + y_{t,s,v} + z_{s,t,v} \leq \sum_{t'\in\Sigma} \left(x_{t',t,v} + x_{t',t,(\pi(v),v)}\right), && \forall s,t\in\Sigma, v\in V(T)\setminus r(T), \\
& y_{s,t,r(T)} + y_{t,s,r(T)} + z_{s,t,r(T)} \leq w_t + \sum_{t'\in\Sigma} x_{t',t,v}, && \forall s,t\in\Sigma, \\
& y_{s,t,v} + y_{t,s,v} + z_{s,t,v} \geq \sum_{s'\in\Sigma} \left(x_{s',s,v} + x_{s',s,(\pi(v),v)}\right) \\
& + \sum_{t'\in\Sigma} \left(x_{t',t,v} + x_{t',t,(\pi(v),v)}\right) - 1, && \forall s,t\in\Sigma, v\in V(T)\setminus r(T), \\
& y_{s,t,r(T)} + y_{t,s,r(T)} + z_{s,t,r(T)} \geq w_s + \sum_{s'\in\Sigma} x_{s',s,v} + w_t + \sum_{t'\in\Sigma} x_{t',t,v} - 1, && \forall s,t\in\Sigma, \\
& y_{s,t,v} \geq \sum_{s'\in\Sigma} x_{s',s,(\pi(v),v)} + \sum_{t'\in\Sigma} x_{t',t,v} - 1, && \forall s,t\in\Sigma, v\in (T)\setminus\{r(T)\}, \\
& y_{s,t,r(T)} \geq w_s + \sum_{t'\in\Sigma} x_{t',t,r(T)} - 1, && \forall s,t\in\Sigma, \\
& \sum_{s'\in\Sigma} \left(x_{s',s,u} + x_{s',s,(\pi(u),u)}\right) \geq \sum_{t\in\Sigma} x_{s,t,(u,v)}, && \forall s\in\Sigma, u\in V(T)\setminus\{r(T)\}, v\in\delta_T(u), \\
& w_s + \sum_{s'\in\Sigma} x_{s',s,r(T)} \geq \sum_{t\in\Sigma} x_{s,t,(u,v)}, && \forall s\in\Sigma, v\in\delta_T(r(T)), \\
& \sum_{s,t\in\Sigma} x_{s,t,(u,v)} = 1, && \forall (u,v)\in E(T), \\
& \sum_{t\in\Sigma} \left(x_{t,s,(\pi_T(v),v)} + x_{t,s,v}\right) = 1, && \forall v\in L(T)\backslash r(T), s\in\hat{\ell}(v), \\
& w_s + \sum_{t\in\Sigma} x_{t,s,r(T)} = 1, && \forall s\in\hat{\ell}(r(T)).
\end{aligned}
$$

D MACH2-Viz

MACH2-Viz contains the following features:

1. Visualization of corresponding nodes and edges between clonal phylogenies and migration graphs.
2. Visualization of the correspondence between input clonal trees and their refined polytomies.
3. Users can view observed locations of input clonal tree nodes, and unobserved clones in the output clonal tree.
4. Users can also compare and examine corresponding components between solutions.

The summary view (see Fig. S3) illustrates the summary graph. This view in the visualizer contains the following features:

1. Users can enforce priors such as known migrations and known absences of migrations to constrain the solution space.
2. If the solution space contains multiple possible primary tumors, the user can constrained to a known primary tumor, or rule out a primary tumor.

We also provide documentation[1] to guide further open-source contributions to the visualizer. Specifically, we offer guidelines for integrating `mach2` results from your research into the `mach2-viz` landing page's table of datasets. This systematic approach ensures a seamless and standardized process for the inclusion of new data.

E Evaluation Metrics

We define three additional evaluation metrics that we use to compare MACH2 with MACHINA on simulated and real data.

E.1 Migration Graph F_1 Score

To compare the solutions inferred by MACH2 and MACHINA with the simulated ground truth instances, we compute the *migration graph F_1 score*, the harmonic mean of the precision and recall of migration edges. More precisely, for an inferred migration graph G' and the corresponding simulated ground truth migration graph G^*, we define recall and precision as the fraction of common edges $E(G') \cap E(G^*)$ in G^* and G', respectively.

$$\text{recall}(G', G^*) = \frac{|E(G') \cap E(G^*)|}{|E(G^*)|} \qquad \text{precision}(G', G^*) = \frac{|E(G') \cap E(G^*)|}{|E(G')|}$$

Note that since migration graphs are multi-edge graphs, the sets $E(G')$ and $E(G^*)$ of edges are also multisets, and thus consider multiplicity while computing intersection and cardinality. From precision and recall, we define the migration graph F_1 score as follows.

$$F_1(G', G^*) = 2 \cdot \frac{\text{recall}(G', G^*) \cdot \text{precision}(G', G^*)}{\text{recall}(G', G^*) + \text{precision}(G', G^*)}$$

Migration graph F_1 score ranges between 0 and 1, while $F_1(G', G^*) = 1$ implying the migration graphs G' and G^* to be exactly the same. Since MACH2 returns a set of solutions unlike MACHINA, we considered the median and minimum migration graph F_1 scores among all the solutions returned by MACH2 (Section 5.1)

[1] `https://elkebir-group.github.io/mach2-docs/`

E.2 Migrating Clone F_1 Score

We define a *migrating clone* to be a refined node u in a solution (T', σ', ℓ') such that there is a migration $(u, v) \in M(T', \sigma', \ell')$. For an inferred set of migrating clones R' and the corresponding simulated ground truth set of migrating clones R^*, we define recall and precision as the fraction of common migrating clones $R' \cap R^*$, respectively.

$$\text{recall}(R', R^*) = \frac{|R' \cap R^*|}{|R^*|} \qquad \text{precision}(R', R^*) = \frac{|R' \cap R^*|}{|R'|}$$

From precision and recall, we define the migrating clone F_1 score as follows.

$$F_1(R', R^*) = 2 \cdot \frac{\text{recall}(R', R^*) \cdot \text{precision}(R', R^*)}{\text{recall}(R', R^*) + \text{precision}(R', R^*)}$$

Migrating clone F_1 score ranges between 0 and 1, while $F_1(R', R^*) = 1$ implying the set of migrating clones R' and R^* to be exactly the same. Since MACH2 returns a set of solutions unlike MACHINA, we considered the median and minimum migrating clone F_1 scores among all the solutions returned by MACH2 (Section 5.1)

E.3 Migration Pattern Totality Ratio (MPTR)

Methods like MACH2 and Metient often return multiple solutions, introducing uncertainty about whether a migration occurred. To quantify this uncertainty, we note that each solution (T', σ', ℓ') induces a set $X(T', \ell')$ of ordered location pairs that correspond to migrations, i.e. $X(T', \ell') = \{(\ell'(u'), \ell(v')) \mid (u', v') \in M(T', \ell')\}$. Given an enumerated space $\mathcal{S}$ of solutions $(T'_1, \sigma'_1, \ell'_1), \ldots, (T'_{|\mathcal{S}|}, \sigma'_{|\mathcal{S}|}, \ell'_{|\mathcal{S}|})$, we define the *migration pattern totality ratio* (MPTR) as the fraction of inferred location pairs that are present in each enumerated solution, i.e. $\text{MPTR}(\mathcal{S}) = |\bigcap_{(T', \sigma', \ell') \in \mathcal{S}} X(T', \ell')| / |\bigcup_{(T', \sigma', \ell') \in \mathcal{S}} X(T', \ell')|$. Thus, an MPTR value of 1 indicates that all inferred solutions agree on the migration pattern (modulo multiplicity), whereas a value of 0 means there is no agreement in the solution space on any specific migration between two locations.