Inferring Temporally Consistent Migration Histories

Mrinmoy Saha Roddur
Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA

Sagi Snir
Department of Evolutionary Biology, University of Haifa, Israel

Mohammed El-Kebir
Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA
Cancer Center at Illinois, University of Illinois at Urbana-Champaign, IL, USA

Abstract

Not only do many biological populations undergo evolution, but population members may also migrate from one location to another. For example, tumor cells may migrate from the primary tumor and seed a new metastasis, and pathogens may migrate from one host to another. One may represent a population's migration history by labeling the vertices of a given phylogeny T with locations such that an edge incident to vertices with distinct locations represents a migration. Additionally, in some biological populations, taxa from distinct lineages may comigrate from one location to another in a single event, a phenomenon known as a comigration. Here, we show that a previous problem statement for inferring migration histories that are parsimonious in terms of migrations and comigrations may lead to temporally inconsistent solutions. To remedy this deficiency, we introduce precise definitions of temporal consistency of comigrations in a phylogeny, leading to three successive problems. First, we formulate the Temporally Consistent Comigrations (TCC) problem to check if a set of comigrations is temporally consistent and provide a linear time algorithm for solving this problem. Second, we formulate the Parsimonious Consistent Comigration (PCC) problem, which aims to find comigrations given a location labeling of a phylogeny. We show that PCC is NP-hard. Third, we formulate the Parsimonious Consistent Comigration History (PCCH) problem, which infers the migration history given a phylogeny and locations of its extant vertices only. We show that PCCH is NP-hard as well. On the positive side, we propose integer linear programming models to solve the PCC and PCCH problems. We apply our approach to real and simulated data.

2012 ACM Subject Classification Applied computing → Computational biology

Keywords and phrases Metastasis, Migration, Integer Linear Programming, Maximum parsimony

Digital Object Identifier 10.4230/LIPIcs.WABI.2023.9

Supplementary Material *Software*: `https://github.com/elkebir-group/PCCH`
archived at `swh:1:dir:f563e890bf28a3f64b5d29b7089358469025fc8c`

Funding *Sagi Snir*: Israel Science Foundation (grant no. ISF 1927/21) and the American/Israeli Binational Science Foundation (grant no. BSF 2021139).
Mohammed El-Kebir: National Science Foundation award number CCF 2046488 as well as funding from the Cancer Center at Illinois.

Acknowledgements This project started as a collaboration at the Computational Genomics Summer Institute 2022.

1 Introduction

Throughout history, various biological populations, ranging from cells and microorganisms to large mammals, have migrated from one place to another. The study of these migrations holds significant importance in various areas of biology and medical science. For instance, understanding the migration history of metastatic cancer can provide insights into the

mechanism of metastasis and aid in the development of novel drugs [7, 11, 12, 26, 31, 33]. Similarly, investigating the transmission of pathogens can help in identifying the source of an outbreak and tracing the patterns of disease spread [3, 10, 13, 14, 32]. Analyzing genomic data is a potential approach to tracing the migration history of a biological population since the migrated subpopulations evolve independently of their counterparts, leading to location-specific genomic differences. One way to do this is by first constructing a rooted phylogeny T, where each vertex v corresponds to a subpopulation with similar genetic makeup, and then labeling each vertex v with their location of origin $\ell(v)$. Therefore each edge (u, v) with distinct labels at its endpoints i.e. $\ell(u) \neq \ell(v)$ corresponds to a migration of subpopulation u from location $\ell(u)$ to location $\ell(v)$ and further evolution to subpopulation v. One challenge with this approach is that although it is possible to know the location of extant subpopulations, the location of extinct subpopulations cannot accurately be known, and so labeling internal vertices of phylogeny T is nontrivial. One parsimony based approach proposed by Slatkin and Madisson [29] to infer internal vertex labeling is to select the labeling that minimizes the number of migrations. Later McPherson et al. [22] used this approach to infer the migration history of cancer cells in metastatic ovarian cancer.

In many evolutionary processes, multiple migrations between the same pair of locations may occur as part of a single event. For instance, in cancer cells from distinct clones may co-migrate as part of a single cluster [1, 2, 5, 6, 8, 11, 17, 20, 21, 35, 36]. Similarly, many pathogens are subject to a weak transmission bottleneck, where multiple variants of the same pathogen are co-transmitted in a single event [27, 28, 30]. MACHINA [11] was the first method to incorporate comigrations in the analysis of metastatic cancer. Specifically, MACHINA defined a comigration to be a set of migrations between the same pair of locations but occurring on different lineages of the tree. In other words, two migrations (u, v) and (u', v') from different lineages belong to the same comigration if $\ell(u) = \ell(u')$ and $\ell(v) = \ell(v')$. Based on this definition, MACHINA extended Slatkin and Madisson [29]'s approach by selecting the location labeling that minimizes the number of migrations, and subsequently the number of comigrations. Following this, another method SharpTNI [27] to infer transmission history was published which uses a similar notion of comigration. One key issue is that this definition of comigration does not adequately capture temporal dependencies between migrations. Time flows from root to leaves in a phylogeny, so if a migration (u, v) occurs before (u', v') in the tree, then all migrations in the comigration containing (u, v) should occur before all migrations in the comigration containing (u', v'). However, the above comigration definition does not enforce this criterion, potentially resulting in temporally inconsistent solutions. In species phylogenetics, similar temporal restrictions arise with lateral gene transfers. Specifically, since gene transfer occurs in co-existing entities, if a transfer occurs from some species A to species B in a species tree, there cannot be another transfer from an ancestor of A to a descendant of B. The temporal consistency of lateral gene transfers has been addressed in studies involving gene tree reconciliation [9, 19, 23, 24, 34], species tree ranking [4], and species tree inference [18].

In this work, we introduce a comigration model that accurately captures both spatial and temporal aspects of simultaneous migrations. To that end, we formulate three new problems. Our first problem, the Temporally Consistent Comigration (TCC) problem aims to assign timestamps to migrations such that migrations in the same comigration have the same timestamp and timestamps are monotonically increasing along the edges of any root-to-leaf path of the tree (Figure 1a). We introduce a linear time algorithm to check if a given set of comigrations is temporally consistent. Our second problem is the Parsimonious Consistent Comigrations (PCC) where, given a rooted tree with locations assigned to all vertices, we seek a minimum set of spatially and temporally consistent comigrations

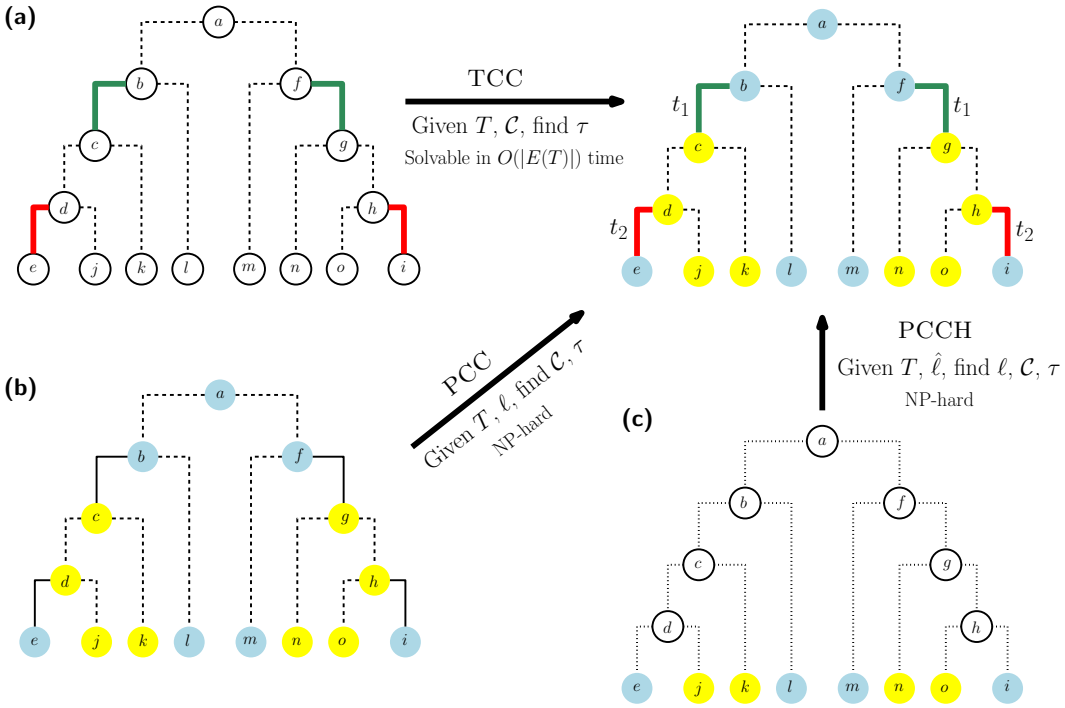


Figure 1 Overview of the three successive problem statements. (a) In the Temporally Consistent Comigrations (TCC) problem, we are given a rooted tree T and a set $\mathcal{C}$ of comigrations (edge colors). We seek a timestamp labeling τ that is temporally consistent with $\mathcal{C}$. Here, timestamp labeling τ for the output tree is indicated by the labels on the edges, where $\tau((b,c)) = \tau((f,g)) = t_1 < t_2 = \tau((d,e)) = \tau((h,i))$, satisfying temporal consistency. (b) In the Parsimonious Consistent Comigrations (PCC) problem, we are no longer given $\mathcal{C}$ only T and a location labeling ℓ (vertex colors). We seek a minimum set $\mathcal{C}$ of comigrations that are spatially and temporally consistent with (T, ℓ). Note that in both TCC and PCC, migrations (solid edges) and non-migrations (dashed edges) are uniquely determined from $\mathcal{C}$ and ℓ, respectively. (c) Finally, in the Parsimonious Consistent Comigration History (PCCH) problem, we are no longer given $\mathcal{C}$ and ℓ, so whether an edge is a migration or not is unknown (dotted edge). Rather we are given T and a leaf location labeling $\hat{\ell}$, seeking a location labeling ℓ inducing a minimum set $|M(T,\ell)|$ of migrations that subsequently admit a smallest set $\mathcal{C}$ of comigrations.

(Figure 1b). We prove this problem to be NP-hard. We then formulate our third problem, Parsimonious Consistent Comigration History (PCCH), where one is given a rooted tree with locations assigned to only the leaves. The goal is to identify a location labeling and comigrations that minimize the number of migrations and subsequently comigrations, while maintaining spatial and temporal consistency (Figure 1c). We show that PCCH is also NP-hard. We formulate integer linear programs for exactly solving PCC and PCCH. We run our methods on real and simulated data, finding that in practice, for small trees with non-complex migration patterns, MACHINA's definition of comigrations is adequate and does not lead to temporal inconsistencies.

2 Problem Statement

Let T be a tree rooted at vertex $r(T)$. As the tree T is rooted, its edges (u, v) are directed such that u is closest to the root $r(T)$ – in this manuscript, we refer to a directed edge or arc as an edge. We denote the vertex set of T by $V(T)$, the edge set by $E(T)$, and the leaf

set by $L(T)$. We refer to root-to-leaf paths as lineages. We write $u \preceq_T v$ to indicate vertex u is an ancestor of vertex v in tree T, i.e. there is a directed path from u to v. Note that $\preceq_T$ is reflexive, i.e. $v \preceq_T v$ for all vertices v. Additionally, we use $\delta(v)$ to denote the set of children of any vertex v. Our goal is to augment T such it allows us to represent a migration history. To that end, following the work of Slatkin and Maddison [29], we let Σ be the set of all locations of origin and define the *location labeling* $\ell : V(T) \to \Sigma$, mapping each vertex with its location of origin as follows.

▶ **Definition 1.** *A* location labeling *is a function $\ell : V(T) \to \Sigma$ that labels the vertices of T with locations from Σ.*

Given a location labeling ℓ of T, we define migrations as edges whose endpoints have different labels.

▶ **Definition 2.** *A* migration *is an edge $(u, v) \in E(T)$ whose endpoints u and v have different locations, i.e. $\ell(u) \neq \ell(v)$. The set of all migrations of T induced by location labeling ℓ is denoted by $M(T, \ell)$.*

In many evolutionary processes, multiple migrations between the pair of locations may occur as part of a single event. To model this, rather than considering migrations in isolation, we wish to partition the set $M(T, \ell)$ of migrations into comigrations $\mathcal{C}$.

▶ **Definition 3.** *A set $\mathcal{C}$ of* comigrations *is a partition of a set $M \subseteq E(T)$ of migrations, i.e. (i) each migration $(u, v) \in M$ occurs in exactly one part and (ii) the union of all parts $C \in \mathcal{C}$ equals M.*

Importantly, not all comigrations $\mathcal{C}$ are valid, as we require all the migrations in each single comigration event to migrate between the same pair of locations at the same time. In other words, we require spatial and temporal consistency defined as follows.

▶ **Definition 4.** *A set $\mathcal{C}$ of comigrations is* spatially consistent *with location labeling ℓ if for all two migrations $(u, v), (u', v')$ in the same part $C \in \mathcal{C}$ it holds that $\ell(u) = \ell(u')$ and $\ell(v) = \ell(v')$.*

To model temporal consistency, we label each migration by a timestamp defined as follows.

▶ **Definition 5.** *A* timestamp labeling *is a function $\tau : M \to \mathbb{N}$ that labels each migration of M with a timestamp.*

We say that comigrations $\mathcal{C}$ are *temporally consistent* if we can assign timestamps to each migration s.t. (i) all edges in the same part occur simultaneously and (ii) time moves forward along the directed edges of the tree.

▶ **Definition 6.** *A set $\mathcal{C}$ of comigrations is* temporally consistent with timestamp labeling τ *provided (i) all pairs $(u, v), (u', v')$ of migrations in the same part $C \in \mathcal{C}$ have the same timestamp, i.e. $\tau((u, v)) = \tau((u, v'))$ and (ii) $\tau((u, v)) < \tau((u', v'))$ for any two migrations $(u, v), (u', v')$ where $v \preceq_T u'$.*

The first problem focuses on determining the chronological order of comigration events. In other words, the goal of the first problem is to identify a timestamp labeling τ that is temporally consistent with a given set $\mathcal{C}$ of comigrations. Formally we define the problem as follows.

▶ **Problem 1** (Temporally Consistent Comigrations (TCC)). *Given a rooted tree T and comigrations $\mathcal{C}$ on migrations $M \subseteq E(T)$, find a timestamp labeling τ s.t. $\mathcal{C}$ is temporally consistent with τ.*

We say that comigrations $\mathcal{C}$ are *temporally consistent* if the above problem has a solution. A variant of the problem is when we are not given the set $\mathcal{C}$ of comigrations but only the location labeling ℓ. The task is to identify the comigration events, i.e. the set of migrations that happened simultaneously. In case there are multiple possible scenarios, we seek the most parsimonious solution, i.e. solution with the fewest comigration events. This leads to the following problem.

▶ **Problem 2** (Parsimonious Consistent Comigrations (PCC)). *Given a rooted tree T with location labeling $\ell : V(T) \to \Sigma$, find comigrations $\mathcal{C}$ of migrations $M(T, \ell)$ s.t. (i) $\mathcal{C}$ is spatially consistent with ℓ, (ii) $\mathcal{C}$ is temporally consistent for some timestamp labeling τ and (iii) the number $|\mathcal{C}|$ of comigrations is minimized.*

In practice, observing the locations of ancestral vertices from data obtained at present is not feasible. So instead of a location labeling on all vertices, we are only given a leaf labeling $\hat{\ell} : L(T) \to \Sigma$ as input, where each leaf $v \in L(T)$ is labeled with a location $\hat{\ell}(v)$ from Σ. Given the leaf labelings, we wish to infer the vertex labeling that corresponds to a most parsimonious solution. Similarly to the problem solved by MACHINA [11], we seek to find the solution that lexicographically minimizes the number of migrations and the number of comigrations.

▶ **Problem 3** (Parsimonious Consistent Comigration History (PCCH)). *Given a rooted tree T with location leaf labeling $\hat{\ell} : L(T) \to \Sigma$, find location labeling ℓ and comigrations $\mathcal{C}$ of $M(T, \ell)$ s.t. (i) $\ell(v) = \hat{\ell}(v)$ for all leaves $v \in L(T)$, (ii) C is spatially consistent with ℓ, (iii) there exist timestamps τ temporally consistent with $\mathcal{C}$ and (iv) the number $|M(T, \ell)|$ of migrations, and subsequently the number $|\mathcal{C}|$ of comigrations is minimized.*

For any tree T with leaf labeling $\hat{\ell} : L(T) \to \Sigma$, it holds that the number $|\mathcal{C}|$ of comigrations is at least $|\Sigma| - 1$ for any set $\mathcal{C}$ of comigrations. To see why, observe that each location is seeded at least once except the location at the root. This lower bound can be achieved trivially by labeling all the internal nodes with the same location. Since minimizing the number of comigrations results in each location being seeded exactly once, we minimize the number of migrations first and then comigrations to allow more complex migration scenarios while retaining simplicity. The key difference between the above problem and the problem solved by MACHINA is that here we include an explicit definition of temporal consistency. We will show in the next section that without this condition migration histories that contain temporal inconsistencies might be inferred.

3 Combinatorial Characterization and Complexity

This section includes the theoretical results on the combinatorial characteristics and complexity of the three discussed problems. Due to space constraints, proofs sketches have been moved to Appendix A.

3.1 Combinatorial Characterization of the TCC Problem

In the TCC problem we are given a set $\mathcal{C}$ of comigrations, partitioning the set M of migrations of a tree T. The task is to identify a timestamp labeling $\tau : M \to \mathbb{N}$ that is temporally consistent with $\mathcal{C}$. To solve this problem, we begin by defining the comigration graph $G_{T,\mathcal{C}}$.

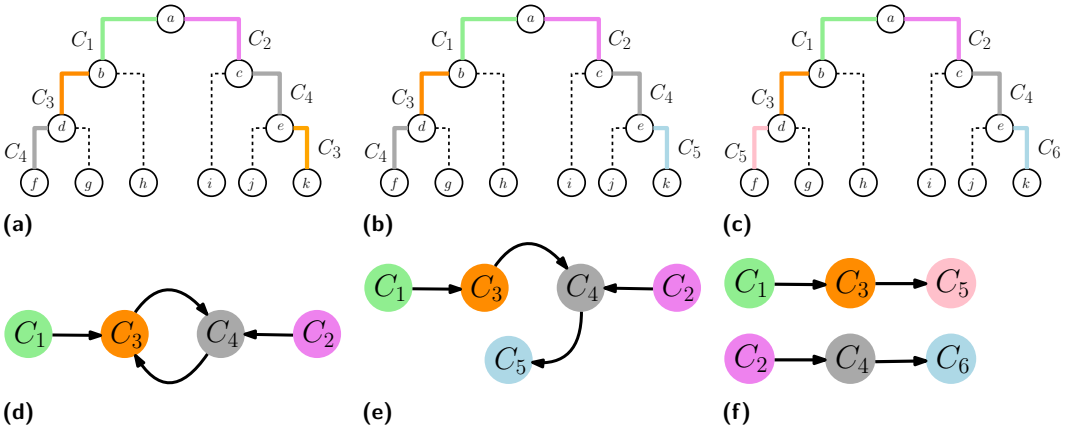


Figure 2 Temporally inconsistent and consistent comigrations with comigration graphs. Illustrated in (a), (b), and (c) are three distinct sets of comigrations within the same tree, where solid edges indicate migrations and dashed edges indicate non-migrations. Edge colors represent the comigrations to which the edges belong, with migrations of the same color belonging to the same comigration. The corresponding comigration graphs for (a), (b), and (c) are shown in (c), (d), and (e). Since the comigration graph illustrated in (d) contains a cycle, the comigrations illustrated in (a) are not temporally consistent. Comigrations corresponding to (b) and (c) are temporally consistent, as the corresponding comigration graphs (e) and (f) are DAG, even though (f) is disconnected.

▶ **Definition 7.** *A* comigration graph $G_{T,\mathcal{C}}$ *for a tree* T *with comigrations* $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$ *is a directed graph with vertices* $V(G_{T,\mathcal{C}}) = \mathcal{C}$ *and a directed edge* $(C_a, C_b) \in E(G_{T,\mathcal{C}})$ *if there exist migrations* $(u_a, v_a) \in C_a$ *and* $(u_b, v_b) \in C_b$ *s.t.* $v_a \preceq_T u_b$ *and* $\mathcal{C}$ *does not contain any other migration on the path from* v_a *to* u_b *in* T*.*

Intuitively, a comigration graph $G_{T,\mathcal{C}}$ orders the comigrations $\mathcal{C}$ by the locations of their corresponding migrations in the tree T. That is, there is an edge (C_a, C_b) in $G_{T,\mathcal{C}}$ if and only if T contains two consecutive migrations on the same lineage, first a migration from C_a followed by a migration from C_b. Note that a comigration graph $G_{T,\mathcal{C}}$ can be disconnected, as shown in Figure 2f. Comigration graphs of sets of comigrations for migrations obtained by a location labeling do not contain self-loops.

▶ **Lemma 8.** *There are no self-loops in the comigration graph* $G_{T,\mathcal{C}}$ *of any set* $\mathcal{C}$ *of comigrations for migrations* $M(T, \ell)$ *induced by location labeling* ℓ *of a tree* T*.*

More generally, comigrations $\mathcal{C}$ admit temporally consistent timestamps if and only if the corresponding comigration $G_{T,\mathcal{C}}$ is a directed acyclic graph (DAG), as we prove in the following proposition.

▶ **Theorem 9.** *There exists a timestamp labeling* τ *that is temporally consistent with comigrations* $\mathcal{C}$ *of a tree* T *if and only if the comigration graph* $G_{T,\mathcal{C}}$ *is a DAG.*

In Section 4.1, we provide an algorithm for solving TCC in $O(|E(T)|)$ time.

3.1.1 MACHINA's Definition of Compatible Comigrations

As we have mentioned earlier, MACHINA [11] was the first method to incorporate comigrations in their problem formulation. Our notion of comigrations is similar to the one introduced in MACHINA [11], but there are significant distinctions. In MACHINA, comigrations $\mathcal{C}$ are

considered valid if for each comigration $C \in \mathcal{C}$, all the migrations belonging to C migrate between the same pair of locations, and no two migrations from C are in the same lineage. In other words, given location labeling ℓ, comigrations $\mathcal{C}$ are valid if they maintain compatibility defined as follows.

▶ **Definition 10** (El-Kebir et al. [11]). *Comigrations $\mathcal{C}$ for migrations $M(T, \ell)$ are* compatible *with location labeling ℓ provided for any two migrations $(u, v), (u', v')$ in the same comigration $C \in \mathcal{C}$ it holds that (i) $\ell(u) = \ell(u')$ and $\ell(v) = \ell(v')$, and (ii) neither $v \preceq_T u'$ nor $v' \preceq_T u$.*

Clearly, compatibility implies spatial consistency. As for the other direction, we have the following lemma relating our notions of spatial and temporal consistency (Definitions 4 and 6, respectively) with compatibility as defined above.

▶ **Lemma 11.** *Comigrations $\mathcal{C}$ for migrations $M(T, \ell)$ that are spatially and temporally consistent with location labeling ℓ of a tree T are also compatible with ℓ.*

The MACHINA paper shows that the minimum number $\gamma(T, \ell)$ of comigrations among all comigrations $\mathcal{C}$ that are compatible with a fixed location labeling ℓ is as follows.

▶ **Lemma 12** (El-Kebir et al. [11]). *The minimum number $\gamma(T, \ell)$ of comigrations among all comigrations compatible with ℓ equals*

$$\gamma(T, \ell) = \sum_{s,t \in \Sigma : s \neq t} \gamma(T, \ell, s, t). \tag{1}$$

where $\gamma(T, \ell, s, t)$ is the maximum number of migrations between locations (s, t) on any root-to-leaf path of T.

The above lemma combined with Lemma 11 leads to the following corollary.

▶ **Corollary 13.** *Comigrations $\mathcal{C}$ that are spatially and temporally consistent with location labeling ℓ of a tree T consist of at least $|\mathcal{C}| \geq \gamma(T, \ell)$ parts.*

While MACHINA only computes the number $\gamma(T, \ell)$ of comigrations and does not explicitly infer corresponding comigrations $\mathcal{C}^*$ s.t. $|\mathcal{C}^*| = \gamma(T, \ell)$, we show here that this can be done using a simple greedy algorithm. Briefly, we initialize $\mathcal{C}^* = \{C_1, \ldots, C_{|M(T,\ell)|}\}$ with each comigration C_i containing migration e_i for all $i \in [|M(T, \ell)|]$. Then, iteratively, we merge two distinct parts C and C' in $\mathcal{C}$ if their comprising migrations are between the same pair of locations and do not occur on the same root-to-leaf path in T. We repeat this procedure until no further merging is possible. Correctness follows from the fact that compatibility is maintained as a loop invariant.

Importantly, while comigrations $\mathcal{C}$ compatible with location labeling ℓ are also spatially consistent with ℓ, temporal consistency is not guaranteed. As an example, consider Figure 3a defining a tree T and location labeling ℓ with locations $\Sigma = \{\text{red}, \text{green}, \text{cyan}, \text{orange}\}$. Tree T and ℓ contain four migrations $(u_{\text{red}}, v_{\text{green}})$, $(u'_{\text{red}}, v'_{\text{green}})$, $(u_{\text{cyan}}, v_{\text{orange}})$ and $(u'_{\text{cyan}}, v'_{\text{orange}})$ s.t. one lineage of T contains the migration $(u_{\text{red}}, v_{\text{green}})$ followed by $(u_{\text{cyan}}, v_{\text{orange}})$ and another distinct lineage contains the migration $(u'_{\text{cyan}}, v'_{\text{orange}})$ followed by $(u'_{\text{red}}, v'_{\text{green}})$. Clearly, $\gamma(T, \ell) = 2$ as no lineage of T contains more than one migration between the same pair of locations. There is a unique set $\mathcal{C}^*$ of comigrations that is compatible with T s.t. $|\mathcal{C}^*| = \gamma(T, \ell) = 2$. That is, $\mathcal{C}^* = \{C_{(\text{red},\text{green})}, C_{(\text{cyan},\text{orange})}\}$ where $C_{(\text{red},\text{green})} = \{(u_{\text{red}}, v_{\text{green}}), (u'_{\text{red}}, v'_{\text{green}})\}$ and $C_{(\text{cyan},\text{orange})} = \{(u_{\text{cyan}}, v_{\text{orange}}), (u'_{\text{cyan}}, v'_{\text{orange}})\}$. Although $\mathcal{C}^*$ is spatially consistent, it is not temporally consistent as can be seen from the cycle in the corresponding migration graph $G_{T,\mathcal{C}^*}$. Indeed, if we assign timestamps τ s.t. migrations

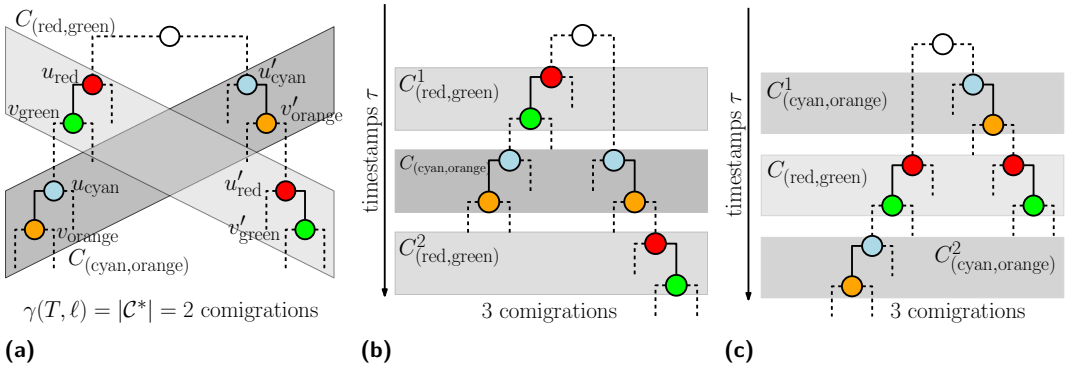


Figure 3 Comigrations inferred by MACHINA [11] might not be temporally consistent. Comigrations $\mathcal{C}^* = \{C_{(\text{red,green})}, C_{(\text{cyan,orange})}\}$ are compatible with the given location labeling ℓ of tree T. Moreover, these comigrations achieve the smallest number $\gamma(T,\ell) = 2$ possible for (T,ℓ). However, the comigration graph $G_{T,\mathcal{C}^*}$ has a cycle between its two vertices $C_{(\text{red,green})}$ and $C_{(\text{cyan,orange})}$. As such, by Theorem 9, $\mathcal{C}^*$ is not temporally consistent. To arrive at temporally consistent comigrations, either (b) $C_{(\text{red,green})}$ or (c) $C_{(\text{cyan,orange})}$ must be split.

in $C_{(\text{red,green})}$ precede $C_{(\text{cyan,orange})}$, we would have a violation of temporal consistency as $(u'_{\text{cyan}}, v'_{\text{orange}}) \preceq_T (u'_{\text{red}}, v'_{\text{green}})$ and yet $\tau((u'_{\text{cyan}}, v'_{\text{orange}})) > \tau((u'_{\text{red}}, v'_{\text{green}}))$. A similar violation would occur when using timestamps s.t. $C_{(\text{cyan,orange})}$ precedes $C_{(\text{red,green})}$. To obtain temporally consistent comigrations, we must break up either $C_{(\text{red,green})}$ (Figure 3b) or $C_{(\text{cyan,orange})}$ (Figure 3c), leading to an additional comigration in either case.

We look deeper into when compatible comigrations $\mathcal{C}$ are temporally consistent. We say a location labeling ℓ results in *reseeding* if there exists a root-to-leaf path in T containing two migrations $(u,v), (u',v')$ labeled as $\ell(u) \neq \ell(v)$, $\ell(u') \neq \ell(v')$ and $\ell(u) = \ell(v')$.

▶ **Proposition 14.** *If a location labeling ℓ of a tree T does not result in reseeding then any set $\mathcal{C}$ of comigrations on $M(T,\ell)$ that is compatible with ℓ is also temporally consistent.*

3.2 NP-Hardness of the PCC Problem

The example shown in Figure 3a and discussed in the previous section shows that not all comigrations $\mathcal{C}$ are temporally consistent, and that to achieve temporal consistency more comigrations than the polynomial-time computable lower bound $\gamma(T,\ell)$ might be needed. In this section, we study the complexity of the PCC problem of finding the smallest set $\mathcal{C}$ of temporally consistent comigrations for migrations $M(T,\ell)$ induced by a location labeling ℓ of a tree T. We have the following hardness result.

▶ **Theorem 15.** *PCC is NP-hard when $|\Sigma| \geq 3$.*

We begin by showing that PCC is NP-hard by reducing it from Shortest Common Supersequence. In Shortest Common Supersequence (SCS) problem, the input is a set $\{S_1, \ldots, S_n\}$ of n sequences, where each sequence S_i is an ordered list $s_{i,1}s_{i,2}\ldots s_{i,|S_i|}$ of symbols from a finite set $\mathcal{S}$. We say sequence Y is a *supersequence* of sequence X if there exists a function $F_{X,Y} : \{1,\ldots,|X|\} \to \{1,\ldots,|Y|\}$ s.t. $F_{X,Y}(i) = j$ if $X_i = Y_j$ and F is a strictly increasing monotone function. In the SCS problem, we seek the shortest sequence S^* s.t. S^* is a supersequence of all input sequences $S_1, \ldots, S_n$. The SCS problem is NP-hard when $|\mathcal{S}| \geq 2$ [25]. We describe a polynomial time reduction from SCS to PCC. To that end, given the input sequences $S_1, \ldots, S_n$, we build a tree T with location set $\Sigma = \mathcal{S} \cup \{\perp\}$ and location labeling $\ell : V(T) \to [\mathcal{S} \cup \{\perp\}]$ in polynomial time. The steps are as follows.

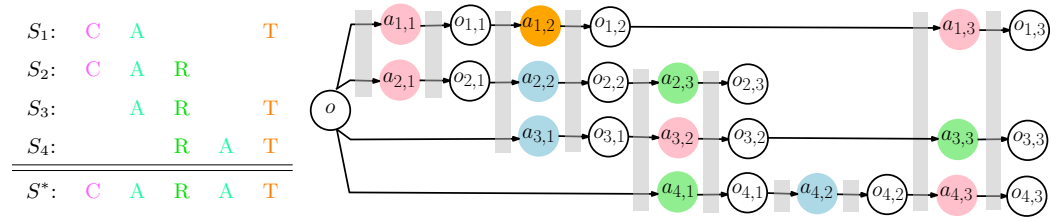


Figure 4 Reduction from Shortest Common Supersequence (SCS) to PCC. (a) An SCS problem instance of $n = 4$ input sequences $\{S_1, \ldots, S_4\}$ with the shortest common supersequence $S^* = s^*_1 \ldots s^*_{m^*}$ of length $m^* = |S^*| = 5$. The solution can be represented as an alignment A, with each column A_p containing pairs (i, j) indicating matched symbols $s_{i,j}$ and s^*_p. (b) The corresponding tree T with location labeling ℓ on $\Sigma = \{\bot, C, A, R, T\}$ is shown. Each vertex $a_{i,j}$ is labeled by location $\ell(v) = s_{i,j}$, with the color matching panel (a). Vertices $o_{i,j}$ are labeled by locations $\ell(v) = \bot$ and are colored white. The corresponding set $\mathcal{C}$ of $2m^* = 2 \cdot 5 = 10$ comigrations is shown using gray boxes, with migrations/edges overlapping a gray box belonging to the same part of $\mathcal{C}$.

1. First we add the root o to tree T. We label the root o with $\ell(o) = \bot$. For convenience, the root o may also be denoted by $o_{i,0}$ for any $1 \leq i \leq n$.
2. For each input sequence S_i, we attach to the root o the path $a_{i,1}, o_{i,1}, \ldots, a_{i,|S_i|}, o_{i,|S_i|}$ of length $2|S_i|$. We refer to vertices $a_{i,j}$ as *a-vertices* and vertices $o_{i,j}$ as *o-vertices*. Note that the edges in the constructed tree are either from an o-vertex to an a-vertex, or from an a-vertex to an o-vertex. As such, we call the former *o-a edges* and the latter *a-o edges*.
3. We set $\ell(a_{i,j}) = s_{i,j}$ and $\ell(o_{i,j}) = \bot$ for each $j \in \{1, \ldots, |S_i|\}$. Since $s_{i,j} \neq \bot$ for all $i \in [n]$ and $j \in \{1, \ldots, |S_i|\}$, all the edges in the tree are migrations.

Since $\Sigma = \mathcal{S} \cup \{\bot\}$ in the PCC instance corresponding to an SCS instance and SCS is NP-hardness when $|\mathcal{S}| \geq 2$, we obtain a lower bound of $|\Sigma| \geq 3$ in Theorem 15. We show an example reduction in Figure 4. Given the constructed tree T with location labeling ℓ, PCC seeks a set $\mathcal{C}^*$ of comigrations that is spatially consistent with ℓ, temporally consistent, and minimizes the number $|\mathcal{C}^*|$ of comigrations. We have the following definition.

▶ **Definition 16.** *A set $\mathcal{C}$ of comigrations for migrations $M(T, \ell) = E(T)$ is* balanced *if $\mathcal{C}$ consists of an even number of parts, half of which comprised of only o-a edges and the other half comprised of only a-o edges.*

▶ **Lemma 17.** *Any optimal set $\mathcal{C}^*$ of comigrations that is spatially and temporally consistent with location labeling ℓ of T is balanced.*

Next, we show there is a mapping between supersequences S of length m and balanced sets $\mathcal{C}$ of $2m$ comigrations that are spatially and temporally consistent with ℓ.

▶ **Lemma 18.** *There exists a common supersequence $S = s_1 \ldots s_m$ of $\{S_1, \ldots, S_n\}$ if and only if there exists a balanced set $\mathcal{C}$ of comigrations with $|\mathcal{C}| = 2m$ parts that is spatially and temporally consistent with location labeling ℓ of T.*

Now we can finally prove the following lemma, from which hardness follows.

▶ **Lemma 19.** *There exists a shortest common supersequence $S^* = s^*_1 \ldots s^*_{m^*}$ of $\{S_1, \ldots, S_n\}$ if and only if there exists a minimum-cardinality set $\mathcal{C}^*$ of comigrations for migrations $M(T, \ell) = E(T)$ that is spatially and temporally consistent with ℓ and has $|\mathcal{C}^*| = 2m^*$ parts.*

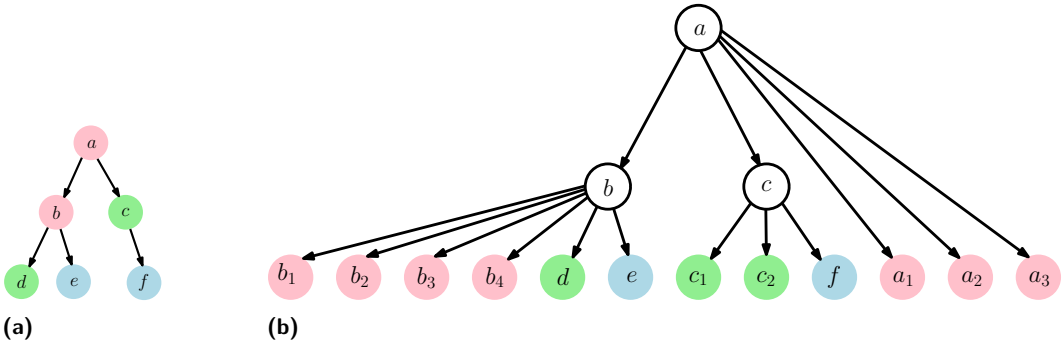


Figure 5 Reduction from PCCH to PCC. (a) An input tree T with vertex labeling ℓ. (b) The corresponding tree T' with leaf labeling $\hat{\ell}'$.

3.3 NP-Hardness of the PCCH Problem

In this subsection, we prove the hardness of PCCH.

▶ **Theorem 20.** *PCCH is NP-hard when* $|\Sigma| \geq 3$.

We prove this by reducing PCC to PCCH in polynomial time. That is, given a tree T with location labeling ℓ, we construct another tree T' with leaf labeling $\hat{\ell}'$. The construction is described below.

1. For each vertex $v \in V(T)$, add vertex v' to $V(T')$.
2. For each edge $(u, v) \in E(T)$, add the edge (u', v') to $E(T')$.
3. For each internal vertex $v \in V(T) \backslash L(T)$ with degree $\deg(v)$ attach $\deg(v) + 1$ leaves $\{v'_1, \ldots, v'_{\deg(v)+1}\}$ to vertex v' of T', labeling each of these leaves with $\ell(v)$, i.e. $\hat{\ell}'(v'_i) = \ell(v)$ for $i \in \{1, \ldots, \deg(v) + 1\}$.
4. For each leaf $v \in L(T)$, retain its label for the corresponding vertex v' in T', i.e. $\hat{\ell}'(v') = \ell(v)$.

Clearly, this reduction takes polynomial time. Since the reduction retains the set Σ of locations of the PCC instance, our hardness result for PCCH has the same bound $|\Sigma| \geq 3$ as in Theorem 15 establishing hardness for PCC. We illustrate the construction with an example in Figure 5. Given the constructed tree T' with leaf labeling $\hat{\ell}'$, PCCH aims to find the location labeling ℓ' as well as spatially and temporally consistent comigrations $\mathcal{C}'$ that result in the minimum number $|M(T', \ell')|$ of migrations and subsequently the minimum number $|\mathcal{C}'|$ of comigrations. As we show in the following lemma, the reduction ensures that an optimal location labeling ℓ' assigns the same locations to internal vertices of T' as location labeling ℓ does to the corresponding internal vertices of T.

▶ **Lemma 21.** *For each vertex* $v \in V(T)$*, an optimal location labeling* ℓ' *of* T' *labels the corresponding vertex* v' *as* $\ell'(v') = \ell(v)$.

The previous lemma means that the number $|M(T', \ell')|$ of migrations is fixed for optimal location labelings ℓ'.

▶ **Corollary 22.** *The number* $|M(T', \ell')|$ *of migrations for an optimal location labeling* ℓ' *of* T' *equals the number* $|M(T, \ell)|$ *of migrations in* T *with location labeling* ℓ.

We now prove the main lemma from which Theorem 20 follows.

▶ **Lemma 23.** *Let (T, ℓ) be a PCC instance with $|M(T, \ell)| = \mu$ and $(T', \hat{\ell}')$ be the corresponding PCCH instance. There exists an optimal solution $\mathcal{C}$ for (T, ℓ) s.t. $|\mathcal{C}| = \gamma$ if and only if there exists an optimal solution $(\ell', \mathcal{C}')$ for $(T', \hat{\ell}')$ s.t. $|M(T', \ell')| = \mu$ and $|\mathcal{C}'| = \gamma$.*

4 Method

In this section, we introduce algorithms to solve the three problems we discussed.

4.1 Linear Time Algorithm for the TCC Problem

Theorem 9 describes a way of solving TCC by computing a topological ordering of the vertices of the given comigration graph $G_{T,\mathcal{C}}$. Using Kahn's algorithm [16], we can obtain the topological ordering in time $O(|V(G_{T,\mathcal{C}})| + |E(G_{T,\mathcal{C}})|)$. As the number $|\mathcal{C}|$ of comigrations is at most the number $|M|$ of migrations, which in turn is at most the number $|E(T)|$ of edges in tree T, we have $|V(G_{T,\mathcal{C}})| = |\mathcal{C}| = O(|E(T)|)$. We bound $|E(G_{T,\mathcal{C}})|$ in the following lemma.

▶ **Lemma 24.** *The number of edges in comigration graph $G_{T,\mathcal{C}}$ is at most the number of edges in T, i.e. $|E(G_{T,\mathcal{C}})| = O(|E(T)|)$.*

Thus, given a comigration graph $G_{T,\mathcal{C}}$, TCC can be solved in $O(|V(G_{T,\mathcal{C}})| + |E(G_{T,\mathcal{C}})|) = O(|E(T)|)$ time. It remains to show how to construct the comigration graph $G_{T,\mathcal{C}}$ itself. Naively, we can check each pair of migrations $(u, v), (u', v') \in M$, and add edge (C_s, C_t) to $G_{T,\mathcal{C}}$ is $(u, v) \in C_s$, $(u', v') \in C_t$, $v \preceq_T u'$ when there is no other migration on the path from v and u'. But this approach is expensive, so we propose a new algorithm that runs in linear time. The recursive algorithm BuildComigrationGraph$(T, M, \mathcal{C}, v)$ takes as input tree T, migration set M, and comigrations $\mathcal{C}$, and a vertex v. It returns two outputs: (i) a comigration graph denoted as $G_{T_v,\mathcal{C}}$, where an edge (C_s, C_t) exists if there are two migrations $(u, v) \in C_s$ and $(u', v') \in C_t$ in the subtree T_v rooted at v, and (ii) a subset $X_v \subseteq \mathcal{C}$ of comigrations s.t. each $C \in X_s$ if C includes a migration (u', v') that is the first migration encountered on a path starting from v. Since $T_{r(T)} = T$, BuildComigrationGraph$(T, M, \mathcal{C}, r(T))$ infers the comigration graph $G_{T,\mathcal{C}}$. The pseudocode is given in Algorithm 1.

▶ **Theorem 25.** *BuildComigrationGraph$(T, M, \mathcal{C}, r(T))$ returns comigration graph $G_{T,\mathcal{C}}$ in $O(|E(T)|)$ time.*

4.2 ILP for the PCC Problem

We solve PCC by formulating an integer linear program that models comigrations $\mathcal{C}$ and timestamp labeling τ for a given tree T and location labeling ℓ, and minimizes over the number of comigrations $|\mathcal{C}|$ while maintaining temporal consistency for some τ. Due to space constraints, we refer the reader to Appendix B.1 for further details.

4.3 ILP for the PCCH Problem

To solve PCCH, we formulate an integer linear program (ILP) that models location labeling ℓ given tree T with leaf labeling $\hat{\ell}$. To do so, we model (i) location labeling, (ii) comigrations characterized by the labels of endpoints and timestamps of the member edges, (iii) assignment of edges to parts, and (iv) additional constraints to break symmetries. We describe each step in detail as follows.

Algorithm 1 BuildComigrationGraph$(T, M, \mathcal{C}, u)$.

Input: Rooted tree T, migrations $M \subseteq E(T)$, comigrations $\mathcal{C}$ and vertex u of T
Output: Comigration graph $G_{T_u,\mathcal{C}}$ for the subtree T_u of T rooted at u and comigrations $\mathcal{C}$ as well as set X_u comprised of parts $C \in \mathcal{C}$ containing a migration (v, w) that is the first migration on the path from u to v

```
if u ∈ L(T) then
    return (G_{T_u,C}, X_u) where V(G_{T_u,C}) = C, E(G_{T_u,C}) = ∅ and X_u = ∅
else
    V(G_{T_u,C}) ← C
    E(G_{T_u,C}) ← ∅
    X_u ← ∅
    foreach child v of u do
        (G_{T_v,C}), X_v) ← BuildComigrationGraph(T, M, C, v)
        E(G_{T_u,C}) ← E(G_{T_u,C}) ∪ E(G_{T_v,C})
        if (u, v) ∈ M then
            Let C_s be the part of C containing (u, v)
            for C_t ∈ X_v do
                E(G_{T_u,C}) ← E(G_{T_u,C}) ∪ {(C_s, C_t)}
            X_u ← X_u ∪ {C_s}
        else
            X_u ← X_u ∪ X_v
    return (G_{T_u,C}, X_u)
```

Location labeling. We introduce binary variables $\Lambda \in \{0,1\}^{|V(T)|\times|\Sigma|}$ to model location labeling ℓ. More specifically, we require $\Lambda_{v,s} = 1$ if $\ell(v) = s$, and $\Lambda(v, s) = 0$ otherwise.

$$\sum_{s\in\Sigma} \Lambda_{v,s} = 1, \qquad \forall v \in V(T).$$

Additionally, for the leaves of T, vertex labeling ℓ should maintain the input leaf labeling $\hat{\ell}$.

$$\Lambda_{v,\hat{\ell}(v)} = 1, \qquad \forall v \in L(T).$$

Timestamp labeling. To efficiently formulate the ILP, we put timestamps on non-migrations too, and include them in comigrations. This does not change the original PCCH algorithm, as we can ignore the timestamps on non-migrations and still get temporal consistency. Similar to our ILP for PCC, we introduce binary variables $\Gamma = \{0,1\}^{|E(T)\times|\Sigma|\times|\Sigma|\times|E(T)|}$ s.t. $\Gamma_{(u,v),s,t,e}$ is 1 if $\ell(u) = s$, $\ell(v) = t$, and $\tau((u, v)) = e$, and $\Gamma_{(u,v),s,t,e} = 0$ otherwise. The maximum number of such unique timestamps is $|E(T)|$, occurring when each edge of the tree is in a distinct comigration. The following three constraints enforce these described conditions.

$$\sum_{t\in\Sigma}\sum_{e\in[|E(T)|]} \Gamma_{(u,v),s,t,e} \le \Lambda_{u,s}, \qquad \forall (u, v) \in E(T), \forall s \in \Sigma,$$

$$\sum_{s\in\Sigma}\sum_{e\in[|E(T)|]} \Gamma_{(u,v),s,t,e} \le \Lambda_{v,t}, \qquad \forall (u, v) \in E(T), \forall t \in \Sigma,$$

$$\sum_{s\in\Sigma}\sum_{t\in\Sigma}\sum_{e\in[|E(T)|]} \Gamma_{(u,v),s,t,e} = 1, \qquad \forall (u, v) \in E(T).$$

To ensure temporal consistency, we require for any two consecutive edges $(u, v), (v, w) \in E(T)$, the timestamp of (u, v) to be smaller than the timestamp of (v, w).

$$\sum_{s\in\Sigma}\sum_{t\in\Sigma}\sum_{e\in[E]}\Gamma_{(u,v),s,t,e} \geq \sum_{s\in\Sigma}\sum_{t\in\Sigma}\sum_{e\in[E]}\Gamma_{(v,w),s,t,e} \quad \forall (u,v),(v,w)\in E(T), \forall E\in[|E(T)|]$$

Comigrations. Just like our ILP model for PCC, we introduce binary variables $\pi \in \{0,1\}^{|E(T)|\times|\Sigma|\times|\Sigma|}$ s.t. $\pi_{e,s,t} = 1$ if for any migration $(u, v) \in C$, it holds that $\ell(u) = s$, $\ell(v) = t$, and $\tau((u, v)) = e$. Again, we require each comigration to have a unique timestamp in this ILP and use the timestamps to identify a specific comigration. We have the following constraint that ensures spatial consistency by enforcing each comigration to be associated with a specific pair of locations.

$$\sum_{s\in\Sigma}\sum_{t\in\Sigma}\pi_{e,s,t},\leq 1 \qquad \forall e\in[|E(T)|].$$

If there is an edge (u, v) with $\ell(u) = s$, $\ell(v) = t$, and $\tau((u, v)) = e$, we force $\pi_{e,s,t}$ to be 1.

$$\Gamma_{(u,v),s,t,e} \leq \pi_{e,s,t}, \qquad \forall (u,v)\in E(T), \forall s,t\in\Sigma, \forall e\in[|E(T)|].$$

Additional constraints. Like the ILP model for PCC, we eliminate some symmetrical solutions by forcing smaller partition numbers to fill up first.

$$\sum_{s\in\Sigma}\sum_{t\in\Sigma}\pi_{e,s,t} \geq \sum_{s\in\Sigma}\sum_{t\in\Sigma}\pi_{e+1,s,t}, \qquad \forall e\in[|E(T)|-1].$$

Optimization function. We can compute the number of migrations from Γ by counting the number of migrations, i.e. edges with different labels at their endpoints. Since we ignore the comigrations with non-migrations, we only count the number of comigrations that contain migrations from π. Thus, given a tree T with location labeling ℓ, we define the objective function as

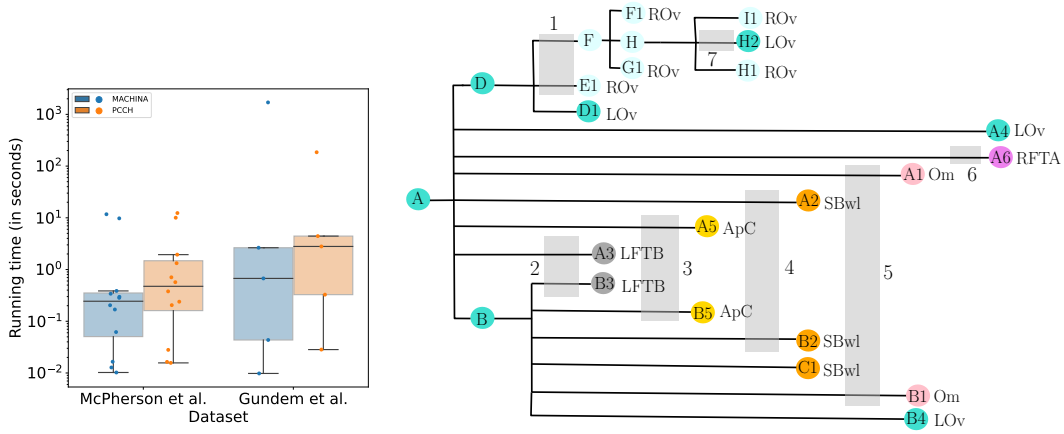
$$\min \sum_{(u,v)\in E(T)}\sum_{s,t\in\Sigma:s\neq t}\sum_{e\in[|E(T)|]}\Gamma_{(u,v),s,t,e} + \frac{1}{|E(T)|}\sum_{e\in[|E(T)|]}\sum_{s,t\in\Sigma:s\neq t}\pi_{e,s,t}.$$

5 Results

In this section, we present a performance comparison between MACHINA and PCCH by running both methods on real (Section 5.1) and simulated data (Section 5.2). 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. The code is available at `https://github.com/elkebir-group/PCCH`.

5.1 Real data

Ovarian cancer. We applied PCCH to infer the migration history of a high-grade serous ovarian cancer dataset by McPherson et al. [22]. The available data contains the phylogenies of seven high-grade serous metastatic ovarian cancer patients. By employing whole genome and single nucleus sequencing, McPherson et al. [22] sequenced 68 tumor samples in total from seven patients including samples from the ovary, omentum, fallopian tube, peritoneal sites, and other distant metastatic sites, and inferred the migration history without considering comigrations. The same dataset was re-analyzed by El-Kebir et al. [11], reporting simpler



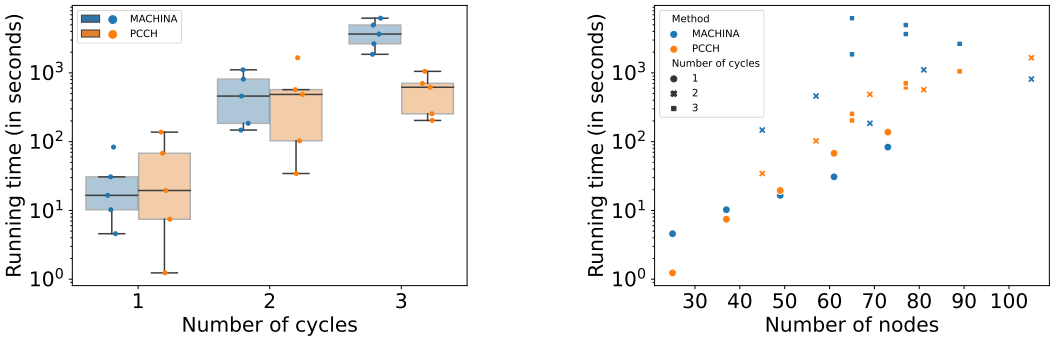
(a) (b)

Figure 6 MACHINA and PCCH results for ovarian [22] and prostate cancer [15] datasets. For all instances, MACHINA and PCCH inferred the same location labeling and the same number of comigrations, meaning that the previously reported MACHINA solutions are indeed temporally consistent. (a) PCCH is slower than MACHINA due to additionally enforcing temporal consistency. (b) PCCH results for ovarian cancer Patient 1. Gray boxes indicate comigrations (additionally labeled by timestamp) and vertex colors indicate location labeling.

migration patterns for some patients using the comigration criterion of MACHINA. For instance, for Patient 7, MACHINA reported that the tumor originated in the left or the right ovary, even though McPherson et al. claimed the right uterosacral ligament to be the primary region. MACHINA found a simpler migration history for Patient 3 without metastasis-to-metastasis migration or multisource seeding, unlike McPherson et al.'s prediction.

For each of the seven patients, we generated the location labeling with timestamps by solving PCCH. We found that PCCH's location labelings perfectly matched those of MACHINA as well as that both methods returned the same number of comigrations, thus showing that MACHINA's solutions were temporally consistent. We show the running time analysis for both PCCH and MACHINA in Figure 6a and Table S1. We found that PCCH generally takes slightly longer to finish (median of 0.474 s vs. 0.244 s for MACHINA). This is expected, as unlike MACHINA, PCCH includes checks for temporal consistency and returns timestamps along with a location labeling. As an example, we show the PCCH output for Patient 1 in Figure 6b with location and timestamp labels. Both MACHINA and PCCH report reseeding in the migration history, which can easily be seen by observing the edges with timestamps 1 and 7. Note that there are other possible timestamp labelings, and PCCH returns only one single solution.

Prostate cancer. We ran PCCH on another dataset by Gundem et al. [15]. The dataset contained matched primary and metastasis samples from five prostate cancer patients. This is another dataset that was previously analyzed by MACHINA [11], where MACHINA could infer alternative migration histories that were simpler and more consistent with the data compared to those reported in the original paper. Though the dataset contained examples of metastasis-to-metastasis spread, MACHINA did not infer reseeding in any of the patients. As such, by Proposition 14, MACHINA's results will be temporally consistent. Indeed, we found that the location labeling from our results from PCCH and the number of comigrations perfectly matched those reported by MACHINA. As shown in Figure 6a and Table S2,



(a) (b)

Figure 7 MACHINA and PCCH results for simulated data. We use colors to differentiate between MACHINA and PCCH. (a) The x-axis corresponds to the number of cycles in the induced comigration graph, where the y-axis stands for running time in seconds. (b) The x-axis represents the number of vertices, and the y-axis stands for running time in seconds. The number of cycles in the induced comigration graph is indicated by marker style.

we observed similar trends for running times (median of 2.795 s for PCCH vs. 0.67 s for MACHINA), although for Patient A22, MACHINA (1702.24 s) took longer than PCCH (185.18 s).

5.2 Simulated data

Results from real data suggest that although it is theoretically possible for MACHINA to return temporally inconsistent solutions, this does not occur in practice. For the same reason, realistic simulation models often fail to generate instances where MACHINA underestimates the number of comigrations. So to assess the performance of PCCH properly, we specifically simulated instances where MACHINA will fail to infer the correct number of comigrations. To be more exact, we sample a comigration graph with $k = \{1, 2, 3\}$ cycles first, and then generate trees with location labeling and comigrations that lead to k cycles in the induced comigration graph (Figure 3a shows an example with $k = 1$ cycle). In our simulated dataset, the cycles in the sampled comigration graph do not share any edges, so the difference between the number of comigrations inferred by PCCH and MACHINA equals the number of cycles. The running time comparison is given in Figure 7 and Table S3. Strikingly, we observed different trends here – MACHINA tends to be slower (median: 459.787 s) than PCCH (median: 203.438 s), especially when the corresponding comigration graph has increasing numbers of cycles (for $k = 3$ cycles, median of 3668.93 and 618.012 seconds for MACHINA and PCCH, respectively).

6 Conclusion

In this paper, we addressed a flaw in the definition of comigration adopted by multiple methods including MACHINA [11]. Specifically, we defined spatial and temporal consistency conditions for comigrations. This led us to formulate three successive problems First, Temporally Consistent Comigration (TCC) asks if a given comigration is temporally consistent, and, if so, returns a certifying timestamp labeling of migrations. We showed that TCC can be solved in linear time. Second, Parsimonious Consistent Comigration (PCC) aims to infer the smallest set of comigrations given a location labeling of both leaf and

internal vertices. We proved the problem is NP-hard, meaning even if we know the location of origin of every vertex and thus every migration, it is still hard to know which migrations occurred simultaneously under a parsimony criterion. Third, we formulated Parsimonious Consistent Comigration History (PCCH), which takes as input a leaf labeling, and infers the location labeling that minimizes the number of migrations, and subsequently the number of comigrations while maintaining spatial and temporal consistency. We showed that PCCH is NP-hard. Additionally, we discussed MACHINA's views on comigrations and its limitations in light of temporal consistency. We also investigated the sufficient conditions for MACHINA to correctly compute the number of comigrations. We compared the performance of PCCH with that of MACHINA on real and simulated data. We observed PCCH to return the same location labeling as MACHINA for all real data instances, which tells us that although it is theoretically possible for MACHINA to fail to compute the correct minimum number of comigrations for some instances, it is unlikely to come across such an instance in practice. Finally, we generated simulated instances where MACHINA fails to determine temporally consistent comigrations.

References

1 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.

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

3 Finlay Campbell, Anne Cori, Neil Ferguson, and Thibaut Jombart. Bayesian inference of transmission chains using timing of symptoms, pathogen genomes and contact data. *PLoS computational biology*, 15(3):e1006930, 2019.

4 Cédric Chauve, Akbar Rafiey, Adrian A Davin, Celine Scornavacca, Philippe Veber, Bastien Boussau, Gergely J Szöllősi, Vincent Daubin, and Eric Tannier. MaxTiC: Fast ranking of a phylogenetic tree by maximum time consistency with lateral gene transfers. *bioRxiv*, page 127548, 2017.

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

6 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.

7 Elizabeth Comen, Larry Norton, and Joan Massague. Clinical implications of cancer self-seeding. *Nature reviews Clinical oncology*, 8(6):369–377, 2011.

8 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.

9 Lawrence A David and Eric J Alm. Rapid evolutionary innovation during an archaean genetic expansion. *Nature*, 469(7328):93–96, 2011.

10 Simon Dellicour, Guy Baele, Gytis Dudas, Nuno R Faria, Oliver G Pybus, Marc A Suchard, Andrew Rambaut, and Philippe Lemey. Phylodynamic assessment of intervention strategies for the West African Ebola virus outbreak. *Nature communications*, 9(1):2222, 2018.

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

12 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.

13 Ousmane Faye, Pierre-Yves Boëlle, Emmanuel Heleze, Oumar Faye, Cheikh Loucoubar, N'Faly Magassouba, Barré Soropogui, Sakoba Keita, Tata Gakou, Lamine Koivogui, et al. Chains of transmission and control of Ebola virus disease in Conakry, Guinea, in 2014: an observational study. *The Lancet Infectious Diseases*, 15(3):320–326, 2015.

14 Neil M Ferguson, Christl A Donnelly, and Roy M Anderson. Transmission intensity and impact of control policies on the foot and mouth epidemic in Great Britain. *Nature*, 413(6855):542–548, 2001.

15 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.

16 Arthur B Kahn. Topological sorting of large networks. *Communications of the ACM*, 5(11):558–562, 1962.

17 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.

18 Manuel Lafond and Marc Hellmuth. Reconstruction of time-consistent species trees. *Algorithms for Molecular Biology*, 15(1):1–27, 2020.

19 Ran Libeskind-Hadas and Michael A Charleston. On the computational complexity of the reticulate cophylogeny reconstruction problem. *Journal of Computational Biology*, 16(1):105–117, 2009.

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

21 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.

22 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.

23 Daniel Merkle and Martin Middendorf. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. *Theory in Biosciences*, 123:277–299, 2005.

24 Nikolai Nøjgaard, Manuela Geiß, Daniel Merkle, Peter F Stadler, Nicolas Wieseke, and Marc Hellmuth. Time-consistent reconciliation maps and forbidden time travel. *Algorithms for Molecular Biology*, 13(1):1–17, 2018.

25 Kari-Jouko Räihä and Esko Ukkonen. The shortest common supersequence problem over binary alphabet is NP-complete. *Theoretical Computer Science*, 16(2):187–198, 1981. doi:10.1016/0304-3975(81)90075-X.

26 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.

27 Palash Sashittal and Mohammed El-Kebir. SharpTNI: Counting and sampling parsimonious transmission networks under a weak bottleneck. *bioRxiv*, page 842237, 2019.

28 Palash Sashittal and Mohammed El-Kebir. Sampling and summarizing transmission trees with multi-strain infections. *Bioinformatics*, 36(Supplement_1):i362–i370, 2020.

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

30 Ashley Sobel Leonard, Daniel B Weissman, Benjamin Greenbaum, Elodie Ghedin, and Katia Koelle. Transmission bottleneck size estimation from pathogen deep-sequencing data, with an application to human influenza A virus. *Journal of virology*, 91(14):e00171–17, 2017.

31 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.

32 Enea Spada, Luciano Sagliocca, John Sourdis, Anna Rosa Garbuglia, Vincenzo Poggi, Carmela De Fusco, and Alfonso Mele. Use of the minimum spanning tree model for molecular epidemiological investigation of a nosocomial outbreak of hepatitis C virus infection. *Journal of clinical microbiology*, 42(9):4230–4236, 2004.

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

34 Ali Tofigh, Michael Hallett, and Jens Lagergren. Simultaneous identification of duplications and lateral gene transfers. *IEEE/ACM transactions on computational biology and bioinformatics*, 8(2):517–535, 2010.

35 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.

36 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.

A Supplementary Proof Sketches

A.1 Combinatorial Characterization of the TCC Problem

▶ **(Main Text) Lemma 8.** *There are no self-loops in the comigration graph $G_{T,\mathcal{C}}$ of any set $\mathcal{C}$ of comigrations for migrations $M(T,\ell)$ induced by location labeling ℓ of a tree T.*

Proof sketch. If there were a self-loop $(C,C) \in E(G_{T,\mathcal{C}})$ then there would be at least one pair $(u,v),(u',v') \in C$ such that there is no migration edge in the path from v to u'. But then $\ell(v) = \ell(u') = \ell(u)$, which means (u,v) is not a migration. ◀

▶ **(Main Text) Theorem 9.** *There exists a timestamp labeling τ that is temporally consistent with comigrations $\mathcal{C}$ of a tree T if and only if the comigration graph $G_{T,\mathcal{C}}$ is a DAG.*

Proof sketch. If $G_{T,\mathcal{C}}$ is a DAG then τ can be constructed by setting $\tau((u,v)) = i$ if (u,v) is in the i-th comigration in the topological ordering of $V(G_{T,\mathcal{C}})$. Also, it can be shown that if there is a cycle $C_i, C_{i+1}, \ldots, C_j, C_i$ in $G_{T,\mathcal{C}}$ then there does not exist a timestamp labeling τ that labels the edges present in $C_i, C_{i+1}, \ldots, C_j$. ◀

A.2 MACHINA's Definition of Compatible Comigrations

▶ **(Main Text) Lemma 11.** *Comigrations $\mathcal{C}$ for migrations $M(T,\ell)$ that are spatially and temporally consistent with location labeling ℓ of a tree T are also compatible with ℓ.*

Proof sketch. The proof follows from the definitions of compatibility, spatial and temporal consistency. ◀

▶ **(Main Text) Proposition 15.** *If a location labeling ℓ of a tree T does not result in reseeding then any set $\mathcal{C}$ of comigrations on $M(T,\ell)$ that is compatible with ℓ is also temporally consistent.*

Proof sketch. It can be shown that there cannot be a cycle in the comigration graph if there is no reseeding. ◀

A.3 NP-Hardness of the PCC Problem

▶ **(Main Text) Lemma 19.** *Any optimal set $\mathcal{C}^*$ of comigrations that is spatially and temporally consistent with location labeling ℓ of T is balanced.*

Proof sketch. It can be shown that if the number of comigrations containing $o - a$ edges is smaller than the number of comigrations containing $a - o$ edges in set $\mathcal{C}^*$, it is possible to construct a set of comigrations $\mathcal{C}'$ with twice the number of comigrations containing $o - a$ edges, indicating that $|\mathcal{C}'| < |\mathcal{C}^*|$, and so $\mathcal{C}^*$ is not optimal. This can be achieved by assigning edges $(a_{i,j}, o_{i,j+1})$ and $(a_{p,q}, o_{p,q+1})$ to the same comigration in $\mathcal{C}'$ if the corresponding $o - a$ edges $(o_{i,j-1}, a_{i,j})$ and $(o_{p,q-1}, a_{p,q})$ belong to the same part in $\mathcal{C}$. A similar argument applies when the number of comigrations containing $o - a$ edges is greater than the number of comigrations containing $a - o$ edges. ◀

▶ **(Main Text) Lemma 20.** *There exists a common supersequence $S = s_1 \dots s_m$ of $\{S_1, \dots, S_n\}$ if and only if there exists a balanced set $\mathcal{C}$ of comigrations with $|\mathcal{C}| = 2m$ parts that is spatially and temporally consistent with location labeling ℓ of T.*

Proof sketch. For each supersequence S of length m, we can construct comigrations $\mathcal{C}$ of size $2m$, and conversely for comigrations $\mathcal{C}$ of size $2m$, there exists a supersequence S of length m. Figure 4 provides an illustrative example of such a relationship between S and $\mathcal{C}$. ◀

▶ **(Main Text) Lemma 21.** *There exists a shortest common supersequence $S^* = s^*_1 \dots s^*_{m^*}$ of $\{S_1, \dots, S_n\}$ if and only if there exists a minimum-cardinality set $\mathcal{C}^*$ of comigrations for migrations $M(T, \ell) = E(T)$ that is spatially and temporally consistent with ℓ and has $|\mathcal{C}^*| = 2m^*$ parts.*

Proof sketch. It can easily be proven using Lemma 18. ◀

A.4 NP-Hardness of the PCCH Problem

▶ **(Main Text) Lemma 23.** *For each vertex $v \in V(T)$, an optimal location labeling ℓ' of T' labels the corresponding vertex v' as $\ell'(v') = \ell(v)$.*

Proof sketch. It can be proven by showing that the number of migrations increases if $\ell'(v') = \ell(v)$. ◀

▶ **(Main Text) Lemma 25.** *Let (T, ℓ) be a PCC instance with $|M(T, \ell)| = \mu$ and $(T', \hat{\ell}')$ be the corresponding PCCH instance. There exists an optimal solution $\mathcal{C}$ for (T, ℓ) s.t. $|\mathcal{C}| = \gamma$ if and only if there exists an optimal solution $(\ell', \mathcal{C}')$ for $(T', \hat{\ell}')$ s.t. $|M(T', \ell')| = \mu$ and $|\mathcal{C}'| = \gamma$.*

Proof sketch. From Lemma 21, it is easy to show that the number of migrations $|M(T', \ell')|$ for PCCH instance (T', ℓ') is μ. Next to complete the proof, it suffices to show two things. First, if $\mathcal{C}$ is an optimal set of comigrations for PCC instance (T, ℓ) then the set $\mathcal{C}'$ of comigrations is optimal for PCCH instance $(T', \hat{\ell}')$ where in $\mathcal{C}'$ a pair of migrations (u', v') and (p', q') belong to the same part if the corresponding migrations (u, v) and (p, q) belong to the same part in $\mathcal{C}$. Second, if $\mathcal{C}'$ is an optimal set of comigrations for PCCH instance

$(T', \hat{\ell}')$ then the set $\mathcal{C}$ of comigrations is optimal for PCC instance (T, ℓ) where in $\mathcal{C}$ a pair of migrations (u, v) and (p, q) belong to the same part if the corresponding migrations (u', v') and (p', q') belong to the same part in $\mathcal{C}$. ◀

A.5 Linear Time Algorithm for the TCC Problem

▶ **(Main Text) Lemma 26.** *The number of edges in comigration graph $G_{T,\mathcal{C}}$ is at most the number of edges in T, i.e. $|E(G_{T,\mathcal{C}})| = O(|E(T)|)$.*

Proof sketch. It can be shown that the number of edges in $G_{T,\mathcal{C}}$ is bounded by the number of migrations, which in turn is bounded by the number of edges. ◀

▶ **(Main Text) Theorem 27.** *BuildComigrationGraph$(T, M, \mathcal{C}, r(T))$ returns comigration graph $G_{T,\mathcal{C}}$ in $O(|E(T)|)$ time.*

Proof sketch. This can be shown by structural induction, proving that the values of $G_{T_u,\mathcal{C}}$ and X_u are properly calculated for each node $u \in V(T)$. ◀

B Supplementary Methods

B.1 ILP for the PCC Problem

We solve PCC by formulating an integer linear program that models comigrations $\mathcal{C}$ and timestamp labeling τ for a given tree T and location labeling ℓ, and minimizes over the number of comigrations $|\mathcal{C}|$ while maintaining temporal consistency for some τ.

Timestamp labeling. We introduce binary variables $\Gamma = \{0,1\}^{|M(T,\ell)\times|M(T,\ell)|}$ to model $\tau : M(T,\ell) \to [\mathcal{N}]$ s.t. $\Gamma_{(u,v),e}$ is 1 if $\tau((u,v)) = e$,and 0 otherwise. Since the number of unique timestamps cannot exceed the number $|M(T,\ell)|$ of migrations, we limit the index e corresponding to timestamps to be at most $|M(T,\ell)|$. Since Γ is modeling a function τ, there cannot be more than one image e for each argument $(u,v) \in M(T,\ell)$:

$$\sum_{e=1}^{|E(T)|} \Gamma_{(u,v),e} = 1, \qquad \forall (u,v) \in M(T,\ell).$$

To ensure temporal consistency, we require $\tau((u,v)) < \tau((u',v')))$ for any two migrations $(u,v),(u',v') \in M(T,\ell)$ where $v \preceq_T u'$, and there is no migration in the path from v to u'. Now if $\tau((u,v)) < \tau((u',v')))$ then for any $E \in \{\tau((u',v')),\ldots,\tau((u,v))\}$ we have $\sum_{e=1}^{E} \Gamma_{(u,v),e} < \sum_{e=1}^{E} \Gamma_{(u',v'),e}$. Conversely, if $\tau((u,v)) = \tau((u',v')))$ then $\sum_{e=1}^{E} \Gamma_{(u,v),e} = \sum_{e=1}^{E} \Gamma_{(u',v'),e}$. We combine these two conditions to form the following constraint.

$$\sum_{e=1}^{E} \Gamma_{(u,v),e} \geq \sum_{e=1}^{E} \Gamma_{(u',v'),e}, \qquad \forall (u,v),(u',v') \in X(T,\ell), E \in [|M(T,\ell)|],$$

where $X(T,\ell)$ consists of all ordered pairs $((u,v),(u',v'))$ of migrations s.t. (i) $(u,v),(u',v') \in M(T,\ell)$, (ii) $v \preceq_T u'$ and (iii) there is no migration in the path from v to u'.

Comigrations. For spatiotemporally consistent comigrations $\mathcal{C}$, each part $C \in \mathcal{C}$ consists of migrations between the same pair of locations indicated by ℓ and have the same timestamp given by a timestamp labeling τ. Thus, to model comigrations, we introduce binary variables $\pi \in \{0,1\}^{|M(T,\ell)|\times|\Sigma|\times|\Sigma|}$, where $\pi_{e,s,t} = 1$ if the migrations in the comigration with timestamp $1 \leq e \leq |M(T,\ell)|$ migrate from $s \in \Sigma$ to $t \in \Sigma$. In other words, $\pi_{e,s,t}$ corresponds to comigration $C \in \mathcal{C}$ if for any migration $(u,v) \in C$ it holds that $\ell(u) = s$, $\ell(v) = t$, and $\tau((u,v)) = e$. Without loss of generality, we require each comigration to have a unique timestamp in this formulation, which we use to identify the comigration. This is enforced by the following constraint.

$$\sum_{s\in\Sigma}\sum_{t\in\Sigma}\pi_{e,s,t} \leq 1, \qquad \forall e \in [|M(T,\ell)|].$$

If there is a migration (u,v) and $\tau((u,v)) = e$, i.e. $\Gamma_{(u,v),e} \leq \pi_{e,\ell(u),\ell(v)} = 1$, then we force $\pi_{e,\ell(u),\ell(v)}$ to be 1.

$$\pi_{e,\ell(u),\ell(v)} \geq \Gamma_{(u,v),e}, \qquad \forall (u,v) \in M(T,\ell), \forall e \in [|M(T,\ell)|].$$

Additional constraints. To increase performance, we eliminate some symmetrical solutions by forcing smaller partition numbers to fill up first.

$$\sum_{s\in\Sigma}\sum_{t\in\Sigma}\pi_{e,s,t} \geq \sum_{s\in\Sigma}\sum_{t\in\Sigma}\pi_{e+1,s,t}, \qquad \forall e \in [|M(T,\ell)| - 1].$$

Optimization function. Since each comigration has a unique timestamp, we can get the total number of comigrations by counting nonzero entries in π.

$$\min \sum_{e\in[|E(T)|]}\sum_{s,t\in\Sigma:s\neq t}\pi_{e,s,t}.$$

Table S1 Detailed results for the ovarian cancer dataset from McPherson et al. [22].

Primary	Patient ID	#vertices	#migrations	#comigrations	Running time (in seconds)	
					MACHINA	PCCH
LOv	Patient 1	24	13	7	0.386	1.941
	Patient 3	36	27	7	9.764	10.042
	Patient 4	18	7	3	0.204	0.378
	Patient 7	19	12	6	0.341	0.569
	Patient 9	9	4	2	0.0128	0.015
ROv	Patient 1	24	13	10	0.284	1.326
	Patient 2	10	2	1	0.01	0.027
	Patient 3	36	27	7	11.725	12.364
	Patient 4	18	6	3	0.0618	0.239
	Patient 7	19	13	6	0.297	0.707
	Patient 9	9	5	2	0.0165	0.0163
	Patient 10	17	6	2	0.169	0.204

Table S2 Detailed results for the prostate cancer dataset from Gundem et al. [15].

Patient ID	#vertices	#migrations	#comigrations	Running time (in seconds)	
				MACHINA	PCCH
A10	15	3	3	0.043	0.327
A22	58	32	14	1702.24	185.18
A29	9	1	1	0.009	0.028
A31	29	13	7	2.64	4.428
A32	27	9	4	0.67	2.795

Table S3 Detailed results for the simulated dataset.

#cycles	#vertices	#migrations	#comigrations		Running time (in seconds)	
			MACHINA	PCCH	MACHINA	PCCH
1	25	6	4	5	4.59	1.24
	37	9	6	7	10.24	7.44
	49	12	8	9	16.56	19.48
	61	15	10	11	30.76	67.71
	73	18	12	13	83.34	137.48
2	45	11	6	8	147.46	34.41
	57	14	8	10	459.79	102.84
	69	17	10	12	184.58	485.99
	81	20	12	14	1107.75	570.13
	105	26	16	18	811.93	1658.93
3	65	16	9	12	6241.98	203.44
	65	16	8	11	1861.78	253.82
	77	19	11	14	4959.63	618.01
	77	19	10	13	3668.93	703.97
	89	22	12	15	2645.30	1052.21