JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 31, Number 5, 2024
© Mary Ann Liebert, Inc.
Pp. 396–415
DOI: 10.1089/cmb.2023.0352

Enforcing Temporal Consistency in Migration History Inference

MRINMOY SAHA RODDUR,[1] SAGI SNIR,[2] and MOHAMMED EL-KEBIR[1,3]

ABSTRACT

In addition to undergoing evolution, members of biological populations may also migrate between locations. Examples include the spread of tumor cells from the primary tumor to distant metastases or the spread of pathogens from one host to another. One may represent migration histories by assigning a location label to each vertex of a given phylogenetic tree such that an edge connecting vertices with distinct locations represents a migration. Some biological populations undergo comigration, a phenomenon where multiple taxa from distinct lineages simultaneously comigrate from one location to another. In this work, 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 phylogenetic tree, leading to three successive problems. First, we formulate the temporally consistent comigration 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 comigrations (PCC) problem, which aims to find comigrations given a location labeling of a phylogenetic tree. We show that PCC is NP-hard. Third, we formulate the parsimonious consistent comigration history (PCCH) problem, which infers the migration history given a phylogenetic tree 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 demonstrate our algorithms on simulated and real data.

Keywords: cancer, metastasis, migration, weak transmission bottleneck.

[1]Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, Illinois, USA.
[2]Department of Evolutionary Biology, University of Haifa, Haifa, Israel.
[3]Cancer Center at Illinois, University of Illinois Urbana-Champaign, Urbana, Illinois, USA.

An earlier version of this article was published in WABI 2023 (doi: 10.4230/LIPIcs.WABI.2023.g).

1. INTRODUCTION

Studying the precise pattern of migration of biological populations 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 (Comen et al., 2011; El-Kebir et al., 2018; Faries et al., 2013; Sanborn et al., 2015; Somarelli et al., 2017; Tabassum and Polyak, 2015). Similarly, investigating the transmission of pathogens can help in identifying the source of an outbreak and tracing the patterns of disease spread (Campbell et al., 2019; Dellicour et al., 2018; Faye et al., 2015; Ferguson et al., 2001; Spada et al., 2004).

To successfully trace the migration history of a biological population, one may analyze genomic data as the migrated subpopulations have evolved independently, resulting in genomic differences that are location specific. More specifically, from the genomic data, one may first construct a phylogenetic tree T with each vertex v corresponding to a subpopulation with similar genetic makeup, and then label each vertex v with their location of origin $\ell(v)$. As such, directed edges (u, v) with distinct labels at their endpoints, that is, $\ell(u) \neq \ell(v)$ indicate subpopulation u migrating from $\ell(u)$ to $\ell(v)$ and evolving into subpopulation v. A key issue is that while locations of extant subpopulations, corresponding to leaves of T, are known, the locations of ancestral subpopulations, corresponding to internal vertices, are typically unknown. Slatkin and Maddison (1989) proposed to use parsimony, inferring an internal vertex labeling that minimizes the number of migrations. Later, McPherson et al. (2016) used the same approach to infer the migration history of cancer cells in metastatic ovarian cancer.

While the approach used by Slatkin and Maddison (1989) and McPherson et al. (2016) considers each migration in isolation, there are evolutionary processes where multiple migrations between the same pair of locations may occur simultaneously. For instance, cancer cells from distinct clones may comigrate as part of a single cluster (Aceto et al., 2014; Birkbak and McGranahan, 2020; Cheung and Ewald, 2016; Cheung et al., 2016; Dadiani et al., 2006; El-Kebir et al., 2018; Kok et al., 2021; Maddipati and Stanger, 2015; Marrinucci et al., 2012; Yamamoto et al., 2023; Yu et al., 2013). Similarly, many pathogens are subject to a weak transmission bottleneck, where multiple variants of the same pathogen are cotransmitted in a single event, including influenza (Sobel Leonard et al., 2017), SARS-CoV-2 (Rambaut et al., 2004; Sashittal and El-Kebir, 2020; Sashittal and El-Kebir, 2019), HIV (Tonkin-Hill et al., 2021), and hepatitis B (Margeridon-Thermet et al., 2009; Wang et al., 2010).

MACHINA (El-Kebir et al., 2018) was the first method to incorporate comigrations in the analysis of metastatic cancer, defining a comigration as a set of migrations that occur on distinct lineages of the tree and are between the same pair of locations. Using this definition, MACHINA extended Slatkin and Maddison (1989)'s approach by choosing the location labeling that first minimized the number of migrations followed by minimizing the number of comigrations. Two other methods, SharpTNI (Sashittal and El-Kebir, 2019) and TiTUS (Sashittal and El-Kebir, 2020), use a similar definition of comigration to infer transmission histories during pathogen outbreaks.

A key problem with the MACHINA definition of comigration is its failure to adequately capture temporal dependencies between migrations. Note that time moves forward along the directed edges of a phylogeny. Therefore, if a migration (u, v) precedes another migration (u', v'), then all migrations in the comigration with (u, v) should occur before those in the comigration with (u', v'). However, MACHINA's comigration definition does not enforce this condition, potentially leading to temporally inconsistent solutions.

In species phylogenetics, similar temporal restrictions arise concerning lateral gene transfers. Specifically, since gene transfer occurs in coexisting entities, if a transfer occurs from a species X to another species Y in a species tree, there cannot be another transfer from an ancestor of X to a descendant of Y. The temporal consistency of lateral gene transfers has been addressed in studies involving gene tree reconciliation (David and Alm, 2011; Libeskind-Hadas and Charleston, 2009; Merkle and Middendorf, 2005; Nøjgaard et al., 2018; Tofigh et al., 2010), species tree ranking (Chauve et al., 2017), and species tree inference (Lafond and Hellmuth, 2020).

Here, we present a new model that enforces spatial and temporal consistency of comigrations as well as three problems that use this new model. First, the temporally consistent comigration (TCC) problem seeks to assign timestamps to migrations such that migrations in the same comigration have the same timestamp and timestamps increase monotonically along the edges of any root-to-leaf path of the tree (Fig. 1a). We present a linear time algorithm to solve TCC. Second, the parsimonious consistent comigrations (PCC)

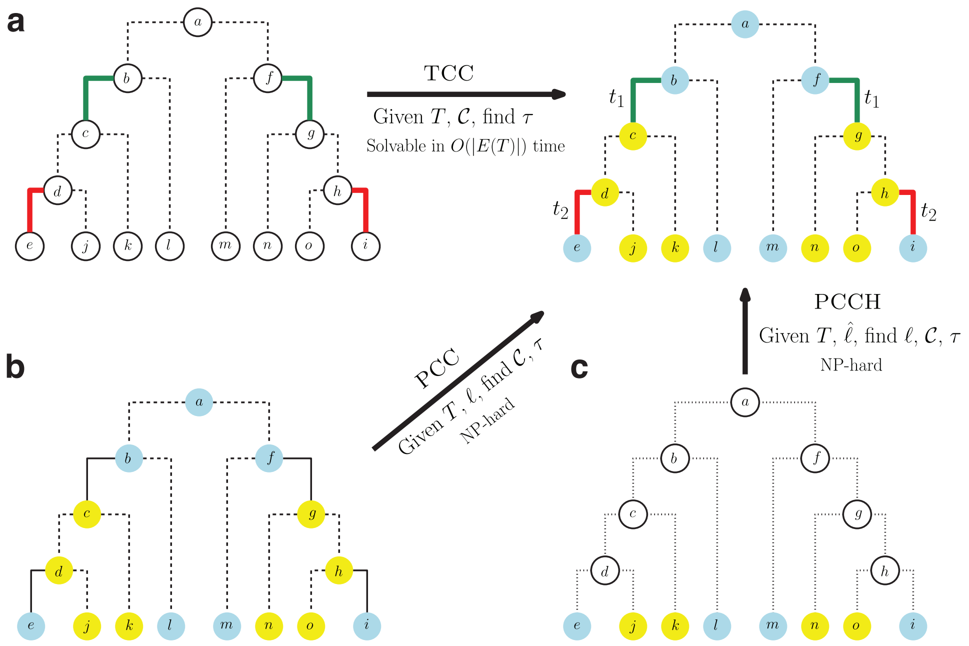
a
TCC
Given T, $\mathcal{C}$, find τ
Solvable in $O(|E(T)|)$ time
t_1
t_2
PCCH
Given T, $\hat{\ell}$, find ℓ, $\mathcal{C}$, τ
NP-hard
b
PCC
Given T, ℓ, find $\mathcal{C}$, τ
NP-hard
c
a
b
f
c
g
d
h
e
j
k
l
m
n
o
i

FIG. 1. Outline of the three problems studied in this article. **(a)** Given a tree T and comigrations $\mathcal{C}$ indicated by edge colors, the TCC problem seeks a timestamp labeling τ that is temporally consistent with $\mathcal{C}$. Here, the timestamps τ are represented by the edge labels, with $t_1 = \tau((f, g)) = \tau((b, c)) < \tau((h, i)) = \tau((d, e)) = t_2$ ensuring temporal consistency. **(b)** Given a location labeling ℓ (vertex colors) of a tree T, the PCC problem seeks a set $\mathcal{C}$ of minimum-cardinality spatiotemporally consistent comigrations. Note that in both TCC and PCC, migrations (indicated by solid edges) and nonmigrations (indicated by dashed edges) are known and uniquely determined by $\mathcal{C}$ and ℓ, respectively. **(c)** Finally, given a tree T and a leaf labeling $\hat{\ell}$, the PCCH problem seeks a location labeling ℓ that admits a minimum-cardinality set $|M(T, \ell)|$ of migrations and subsequently induces the smallest, spatiotemporally consistent set $\mathcal{C}$ of comigrations. PCC, parsimonious consistent comigrations; PCCH, parsimonious consistent comigration history; TCC, temporally consistent comigrations.

problem seeks a minimum-cardinality set of spatially and TCC given a rooted tree with locations assigned to all vertices (Fig. 1b). We prove that this problem is NP-hard. Third, we formulate the parsimonious consistent comigration history (PCCH) problem, where, given a rooted tree with locations assigned to only the leaves, we seek a location labeling and comigrations that minimize the number of migrations and subsequently comigrations, while maintaining spatial and temporal consistency (Fig. 1c). We prove that PCCH is also NP-hard.

We formulate integer linear programs (ILPs) for exactly solving PCC and PCCH. We introduce a workflow for checking MACHINA migration histories for temporally consistency, and, if necessary, correcting them using the problems and algorithms introduced in this article. On simulated data, we find that MACHINA may fail to return temporally consistent solutions. On real data of metastatic cancers with relatively small phylogenetic trees, we find that MACHINA returned temporally consistent solutions. In summary, this work addresses a deficiency in a previous mathematical model of comigration, providing precise definitions and conditions for temporal consistency.

2. PROBLEM STATEMENT

We consider directed trees T rooted at a vertex $r(T)$. We use the term edge to refer to a directed edge or arc, denoted as the pair (u, v) where the vertex u is closest to the root $r(T)$. Vertices of T are denoted by $V(T)$, edges by $E(T)$, and leaves by $L(T)$. We use the term lineage to refer to root-to-leaf paths of T. To indicate that vertex u is an ancestor of vertex v, that is, there is a directed path from u to v, we write $u \preceq_T v$. We note that it holds that $v \preceq_T v$ for all vertices v, that is, the relation $\preceq_T$ is reflexive. We denote the set of children of any vertex v by $\delta(v)$. To represent migration histories, we follow the work of Slatkin and

Maddison (1989) and let Σ be the set of all locations of origin and define a labeling $\ell : V(T) \rightarrow \Sigma$ of the vertices of T by locations Σ, called the *location labeling*, as follows.

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

Migrations are edges of T whose endpoints are assigned different locations by ℓ.

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

We say a location s is *seeding* a location t if there exists a migration $(u, v) \in M(T, \ell)$ such that $\ell(u) = s$, $\ell(v) = t$, and $s \neq t$. As mentioned previously, some evolutionary process allow for multiple migrations between the same pair of locations to occur in a single event. Thus, we wish to partition the set $M(T, \ell)$ of migrations into set $\mathcal{C}$ of comigrations rather than considering each migration in isolation.

Definition 3. A set $\mathcal{C}$ of comigrations is a partition of a set $M \subseteq E(T)$ of migrations, that is, (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.

For comigrations $\mathcal{C}$ to be valid, all the migrations belonging to the same comigration needs to migrate between the same pair of locations at the same time. To that end, we define spatial and temporal consistency 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 first introduce a *timestamp labeling* that labels each migration by a timestamp defined as follows.

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

We now define temporal consistency as follows.

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, that is, $\tau((u, v)) = \tau((u, v'))$ and (ii) $\tau((u, v)) < \tau((u', v'))$ for any two migrations $(u, v), (u', v')$ where $v \preceq_T u'$.

For the first problem, we focus on finding the chronological order of comigrations. That is, given a set $\mathcal{C}$ of comigrations, we wish to identify a timestamp labeling τ with which $\mathcal{C}$ is temporally consistent.

Problem 1 (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 corresponding TCC problem instance has a solution.

Next, we consider the problem where we are no longer given the set $\mathcal{C}$ of comigrations but only the location labeling ℓ and seek to identify temporally consistent comigrations $\mathcal{C}$. As there may be multiple possible scenarios, we seek the most parsimonious solution, that is, the solution with the fewest comigrations, leading to the following problem.

Problem 2 (PCC). *Given a rooted tree T with location labeling $\ell : V(T) \rightarrow \Sigma$, 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.*

We note that in practice, we are only given a leaf labeling $\hat{\ell} : L(T) \rightarrow \Sigma$ as input, where each leaf $v \in L(T)$ is labeled with a location $\hat{\ell}(v)$ from Σ, rather than a location labeling that labels all vertices of T. In the third problem, we wish to infer the vertex labeling that corresponds to a most parsimonious solution for the given leaf labeling. Similarly to the problem solved by MACHINA (El-Kebir et al., 2018), we seek to find the solution that lexicographically minimizes the number of migrations and the number of comigrations. The key difference between the PCCH problem posed below and the previous problem solved by MACHINA is that here we explicitly enforce temporal consistency.

Problem 3 (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.*

To understand why we chose this particular ordering of the two objectives, note that there is a trade-off between the number of migrations and comigrations, where minimizing one objective comes at the expense of the other. Assuming that the location leaf labeling $\hat{\ell}$ is injective, that is, for each location s in Σ, there exists at least one leaf v such that $\hat{\ell}(v) = s$, it holds that the number $|\mathcal{C}|$ of comigrations is at least $|\Sigma| - 1$ for any location labeling ℓ and corresponding set $\mathcal{C}$ of comigrations subject to conditions (i) and (ii) of the PCCH problem. To see why, observe that each location must be seeded or migrated into at least once except or the location at the root $r(T)$. In other words, for each of the $s \in |\Sigma| \setminus \{\ell(r(T))\}$ locations, there is at least one migration $(u, v) \in M(T, \ell)$ such that $\ell(u) \neq s$ and $\ell(v) = s$. There always exists a (temporally-consistent) location labeling with $|\Sigma| - 1$ comigrations, for example, labeling all the internal vertices with the same location.

Location labelings with $|\Sigma| - 1$ comigrations correspond to tree-like migration histories, with each location not equal to $\ell(r(T))$ seeded by exactly one other location. To allow for more complex migration scenarios, we follow the problem statement introduced in El-Kebir et al. (2018) and minimize the number of migrations first and then comigrations. Note that the problem with the two objectives reversed, that is, minimizing comigrations first followed by migrations, was previously considered and shown to be NP-hard (El-Kebir, 2018).

3. COMBINATORIAL CHARACTERIZATION AND COMPLEXITY

This section includes the theoretical results on the combinatorial characteristics and complexity of the three discussed problems. The proofs have been moved to Appendix A (Supplementary Data) because of space constraints.

3.1. Combinatorial characterization of the TCC problem

To solve the TCC problem, we define the comigration graph $G_{T,\mathcal{C}}$, which is obtained from a tree T with comigrations $\mathcal{C}$ as follows.

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.

A comigration graph $G_{T,\mathcal{C}}$ seeks to order comigrations $\mathcal{C}$ by the placement of their corresponding migrations in T. More specifically, $G_{T,\mathcal{C}}$ contains an edge (C_a, C_b) if and only if a migration from C_a immediately precedes a migration from C_b on the same root-to-leaf path in T. Note $G_{T,\mathcal{C}}$ need not be connected (Fig. 2c). On the contrary, comigration graphs for migrations obtained by a location labeling do not contain self-loops.

Lemma 1. *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.*

We have the following important theorem, stating that comigrations $\mathcal{C}$ admit temporally consistent timestamps if and only if $G_{T,\mathcal{C}}$ is a directed acyclic graph (DAG)—see Figure 2.

Theorem 1. *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.*

We show how to solve TCC in $O(|E(T)|)$ time in Section 4.1.

3.2. Relationship to MACHINA's comigrations

As we have mentioned earlier, MACHINA (El-Kebir et al., 2018) was the first method to incorporate comigrations in its problem formulation. Our notion of comigrations is similar to the one introduced in MACHINA (El-Kebir et al., 2018), but there are significant distinctions. MACHINA requires comigrations $\mathcal{C}$ such that for each comigration $C \in \mathcal{C}$, all the migrations belonging to C migrate between the same pair of

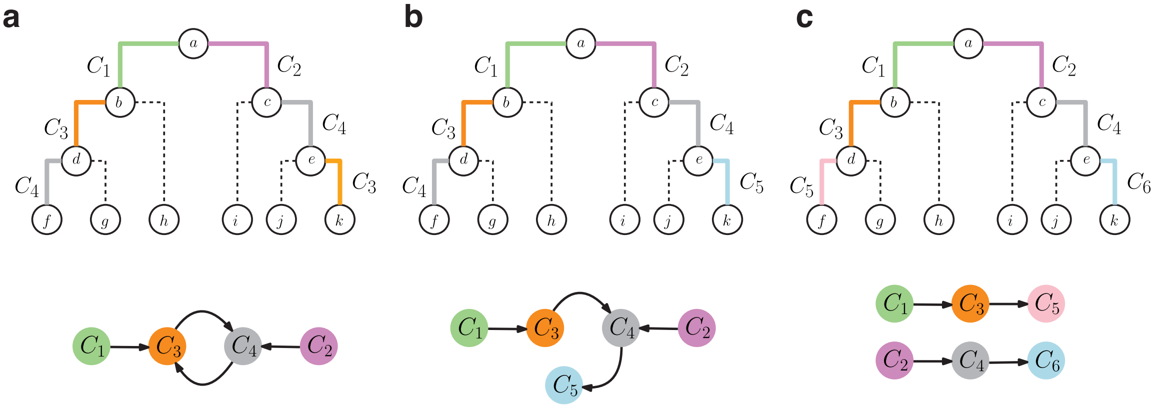

FIG. 2. Temporally inconsistent and consistent comigrations with comigration graphs. **(a–c)** Three distinct sets of comigrations (edge colors) in the same tree with migrations (solid edges) and nonmigrations (dashed edges), resulting in different comigration graphs. **(a)** The comigration graph contains a directed cycle between C_3 and C_4, and therefore the corresponding set of comigrations is temporally inconsistent. **(b, c)** The comigration graphs are DAGs and, therefore, the corresponding sets of comigrations are temporally consistent. DAG, directed acyclic graph.

locations, and no two migrations from C are in the same root-to-leaf path. In other words, MACHINA considers comigrations $\mathcal{C}$ to be valid if they maintain *compatibility* defined as follows.

Definition 8 (El-Kebir et al., 2018). 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 $v' \preceq_T u$.

The minimum number $\gamma(T, \ell)$ of comigrations among all comigrations $\mathcal{C}$ that are compatible with a fixed location labeling ℓ can be computed as follows.

Lemma 2 (El-Kebir et al., 2018). *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.

While comigrations $\mathcal{C}$ compatible with location labeling ℓ are clearly spatially consistent, they may not be temporally consistent. We give one example in Figure 3 where the comigrations are compatible with the location labeling ℓ but not temporally consistent. The following lemma relates our notions of spatial and temporal consistency (Definitions 4 and 6, respectively) with compatibility (Definition 8).

Lemma 3. *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 following corollary directly follows from Lemma 2 and Lemma 3.

Corollary 1. *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.*

Note that MACHINA only computes the number $\gamma(T, \ell)$ of comigrations and does not explicitly infer the corresponding comigrations $\mathcal{C}^*$ s.t. $|\mathcal{C}^*| = \gamma(T, \ell)$. We present a simple greedy algorithm, denoted as Greedy-Comigrations (T, ℓ), to infer $\mathcal{C}^*$. In brief, the algorithm starts with $\mathcal{C}^* = \{C_1, \ldots, C_{|M(T, \ell)|}\}$, where each comigration $C \in \mathcal{C}^*$ contains exactly one unique migration from $M(T, \ell)$. Then in each iteration, two distinct parts C and C' are merged in $\mathcal{C}^*$ if all the migrations from C and C' are between the same pair of locations and exist in distinct root-to-leaf paths in T. The algorithm continues until no more comigration pairs can be merged. We refer to Algorithm 2 in Appendix B.1 for pseudocode more formally

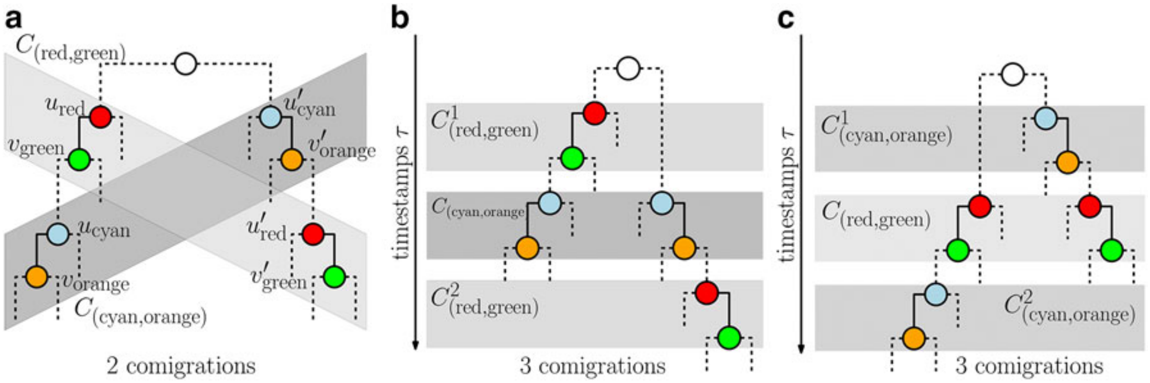

FIG. 3. Comigrations inferred by MACHINA (El-Kebir et al., 2018) might not be temporally consistent. **(a)** Given the tree T and location labeling ℓ with locations Σ indicated by colors $\Sigma = \{\text{red, green, cyan, orange}\}$, comigrations $\{C_{(\text{red, green})}, C_{(\text{cyan, orange})}\}$ indicated by gray boxes are compatible with ℓ. However, assigning timestamps τ such that $\tau((u'_{\text{cyan}}, v'_{\text{orange}})) > \tau((u'_{\text{red}}, v'_{\text{green}}))$ violates temporal consistency as $(u'_{\text{cyan}}, v'_{\text{orange}}) \preceq_T (u'_{\text{red}}, v'_{\text{green}})$. A similar violation happens if the timestamp of $C_{(\text{cyan, orange})}$ precedes that of $C_{(\text{red, green})}$ instead. To get comigrations that are temporally consistent, we must break up either **(b)** $C_{(\text{red, green})}$ or **(c)** $C_{(\text{cyan, orange})}$, leading to an additional comigration in either case.

describing this algorithm. Note that the algorithm maintains compatibility as a loop invariant, which ensures correctness.

Lemma 4. *For any rooted tree T and location labeling ℓ,* Greedy-Comigrations (T, ℓ) *infers comigrations $\mathcal{C}^*$ for $M(T, \ell)$ s.t. (i) $\mathcal{C}^*$ is compatible with ℓ and (ii) $|\mathcal{C}^*| = \gamma(T, \ell)$.*

Note that the greedy approach is not guaranteed to output comigrations that are both compatible with location labeling ℓ and temporally consistent, even if there exist compatible comigrations $\mathcal{C}$ for a given tree T and location labeling ℓ such that $|\mathcal{C}| = \gamma(T, \ell)$. One such example is discussed in Appendix B.1 and Supplementary Figure S1.

Finally, we explore a sufficient condition under which compatible comigrations $\mathcal{C}$ exhibit temporal consistency. We say a location labeling ℓ results in *reseeding* if there exists k distinct migrations $(u_1, v_1), \ldots, (u_k, v_k)$ such that $\ell(v_i) = \ell(u_{i+1})$ for any $1 \leq i \leq k$ and $\ell(u_1) = \ell(v_k)$. In other words, the directed multigraph formed by vertices Σ and containing a directed edge $[\ell(u), \ell(v)]$ for each migration $(u, v) \in M(T, \ell)$—called migration graph in El-Kebir et al. (2018)—is acyclic. We show that comigrations $\mathcal{C}$ compatible with a location labeling ℓ that does not result in reseeding are temporally consistent in the following proposition.

Proposition 1. *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.*

This means that versions of MACHINA that restrict location labeling ℓ to not have reseeding, including versions that only support tree-like migration patterns with $|\Sigma| - 1$ comigrations (El-Kebir, 2018), return temporally consistent solutions, although the solution may be suboptimal for the original unrestricted problem. Similarly, TiTUS (Sashittal and El-Kebir, 2020), which considers timed phylogenetic trees and imposes tree-like migration constraints (i.e., each location is seeded by at most one other location), will result in temporally consistent solutions.

In conclusion, MACHINA does not guarantee temporal consistency unless the inferred location labeling is reseeding-free. We will make use of this when developing a workflow for solving the PCCH problem in Section 4.4.

3.3. NP-hardness of the PCC problem

The example in Figure 3 and Lemma 1 demonstrates that the smallest set $\mathcal{C}$ of TCC can have more comigrations than the polynomial-time computable lower bound $\gamma(T, \ell)$. In this section, we explore the complexity of PCC, which seeks 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 2. *PCC is NP-hard when* $|\Sigma| \geq 3$.

We prove this by a reduction from shortest common supersequence (SCS) in polynomial time. The SCS problem takes as input 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 $F_{X,Y} : \{1, \ldots, |X|\} \rightarrow \{1, \ldots, |Y|\}$ such that $F_{X,Y}(i)=j$ if $X_i = Y_j$ and F is a strictly increasing monotone function. The goal of the SCS problem is to find the shortest sequence S^* such that S^* is a supersequence of all input sequences $S_1, \ldots, S_n$. The SCS problem is NP-hard when $|\mathcal{S}| \geq 2$ (Räihä and Ukkonen, 1981). We describe a polynomial time reduction from SCS to PCC. To that end, we build a tree T with location set $\Sigma = \mathcal{S} \cup \{\perp\}$ and location labeling $\ell : V(T) \rightarrow [\mathcal{S} \cup \{\perp\}]$ given the input sequences $S_1, \ldots, S_n$ in polynomial time. The construction is described below.

1. Add the root o to the empty tree T and set the label of root o to be $\ell(o) = \perp$. For convenience, The root o may also be represented as $o_{i,0}$ for any $1 \leq i \leq n$.
2. For each input sequence S_i, attach the path $a_{i,1},\ o_{i,1},\ \ldots,\ a_{i,|S_i|},\ o_{i,|S_i|}$ of length $2|S_i|$ to the root o. Vertices $a_{i,j}$ are referred to as a-vertices, while vertices $o_{i,j}$ are referred to as o-vertices. By construction, the edges in the tree T are either from an o-vertex to an a-vertex or from an a-vertex to an o-vertex. These are, respectively, called o-a edges and a-o edges.
3. Label each a-vertex $a_{i,j}$ with $\ell(a_{i,j}) = s_{i,j}$ and each o-vertex $o_{i,j}$ with $\ell(o_{i,j}) = \perp$. Since $s_{i,j} \neq \perp$ for all $i \in [n]$ and $j \in \{1, \ldots, |S_i|\}$, each edge of the tree T is a migration.

The lower bound of $|\Sigma| \geq 3$ in Theorem 2 is established by combining the facts that $\Sigma = \mathcal{S} \cup \{\perp\}$ in the PCC instance corresponding to an SCS instance with set $\mathcal{S}$ of symbols, and SCS is NP-hard when $|\mathcal{S}| \geq 2$. Figure 4 shows an example reduction.

In the following, let (T, ℓ) be the PCC instance obtained from SCS instance $\{S_1, \ldots, S_n\}$. Moreover, we denote with $\mathcal{C}^*$ any optimal solution of the PCC instance, that is, $\mathcal{C}^*$ is a set of comigrations that is spatially consistent with ℓ, temporally consistent, and minimizes the number $|\mathcal{C}^*|$ of comigrations. We have the following definition.

Definition 9. 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 only o-a edges and the other half comprised only a-o edges.

Lemma 5. *Any optimal set* $\mathcal{C}^*$ *of comigrations that is spatially and temporally consistent with location labeling* ℓ *of* T *is balanced.*

Next, we show that there exists a mapping between supersequences S of length m and balanced sets $\mathcal{C}$ of $2m$ spatiotemporally consistent comigrations.

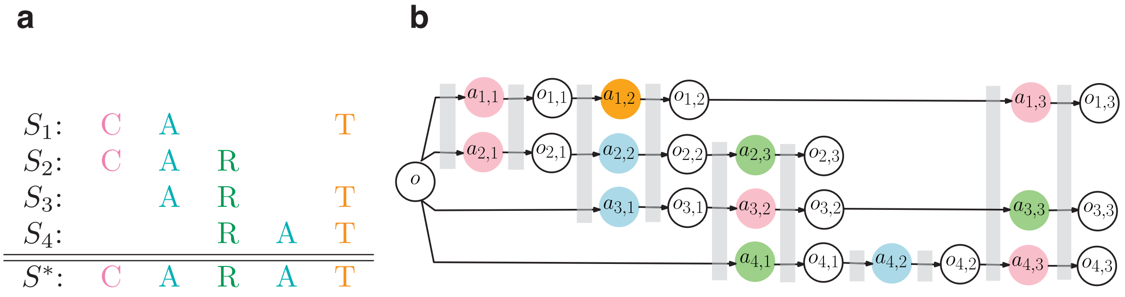

FIG. 4. Reduction from SCS to PCC. **(a)** Given an SCS problem instance with $n = 4$ sequences S_1, S_2, S_3, S_4, we have the SCS $S^* = s_1^* \ldots s_{m^*}^*$ of length $m^* = |S^*| = 5$. The solution is illustrated as an alignment such that $s_{i,j}$, the jth character of sequence i, is in column p if $s_{i,j}$ matches with the p-th character s_p^* of S^*. **(b)** The corresponding tree T with location labeling ℓ on $\Sigma = \{\perp, C, A, R, T\}$ is shown. Each a-vertex $a_{i,j}$ is labeled by location $\ell(a_{i,j}) = s_{i,j}$, with the color matching **(a)**, and each o-vertices $o_{i,j}$ is labeled by locations $\ell(o_{i,j}) = \perp$ and are colored white. The corresponding set $\mathcal{C}$ of $2m^* = 2 \cdot 5 = 10$ comigrations is indicated by gray boxes, with migrations/edges overlapping a gray box belonging to the same part of $\mathcal{C}$. SCS, shortest common supersequence.

Lemma 6. *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.

Finally, we prove the following lemma from which Theorem 2 follows.

Lemma 7. *There exists a SCS* $S^* = s_1^* \ldots s_{m^*}^*$ *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.*

3.4. NP-hardness of the PCCH problem

In this subsection, we prove PCCH to be NP-hard.

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

We show that PCCH is NP-hard by reduction from PCC. To that end, given a tree T with location labeling ℓ, we construct another tree T' with leaf labeling $\hat{\ell}'$. The steps are as follows.

1. For every vertex $v \in V(T)$, add a vertex v' to $V(T')$.
2. For every edge $(u, v) \in E(T)$, add an edge (u', v') to $E(T')$.
3. For every leaf $v \in L(T)$, keep its label $\ell(v)$ for the corresponding vertex v' in T', that is, $\hat{\ell}'(v') = \ell(v)$.
4. 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 $\ell(v)$, that is, $\hat{\ell}'(v'_i) = \ell(v)$ for $i \in \{1, \ldots, \deg(v) + 1\}$.

Clearly, the reduction described above takes polynomial time. Note that the set Σ of locations is the same for both the PCC instance and the corresponding PCCH instance. Therefore, our hardness result for PCCH has the same bound $|\Sigma| \geq 3$ as in Theorem 2 establishing hardness for PCC. We give an example construction in Supplementary Figure S2.

Given the constructed tree T' with leaf labeling $\hat{\ell}'$ from PCC instance (T, ℓ), PCCH aims to find the location labeling ℓ' as well as spatially and TCC $\mathcal{C}'$ that result in the minimum number $|M(T', \ell')|$ of migrations and subsequently the minimum number $|\mathcal{C}'|$ of comigrations. 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 v of T, as we show in the following lemma.

Lemma 8. *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 2. *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.

Finally, we prove the main lemma from which hardness follows.

Lemma 9. *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. METHODS

In this section, we introduce algorithms to solve the three problems we discussed, and also introduce a workflow for inferring a temporally consistent migration history from input trees with leaf labeling.

4.1. Linear time algorithm for the TCC problem

The proof of Theorem 1 describes a way of solving TCC by computing a topological ordering of the vertices of the given comigration graph $G_{T,\mathcal{C}}$. A *topological ordering* is a linear ordering of the comigration graph's vertices such that for every edge (u, v) vertex u comes before v in the ordering; such an ordering exists if and only if $G_{T,\mathcal{C}}$ is a DAG. Given a topological ordering $t : \mathcal{C} \to \{1, \ldots, |\mathcal{C}|\}$ of the vertices $V(G_{T,\mathcal{C}}) = \mathcal{C}$, we can obtain timestamps $\tau : M \to \mathbb{N}$ by setting $\tau((u, v)) = t(C)$ if the migration $(u, v) \in C$ where C is a comigration in the set $\mathcal{C}$ of comigrations. Using Kahn's algorithm (Kahn, 1962), we can obtain the topological ordering in time $O(|V(G_{T,\mathcal{C}})| + |E(G_{T,\mathcal{C}})|)$. Since the number $|\mathcal{C}|$ of comigrations can be at most the number $|M|$ of migrations, which in turn can be at most the number $|E(T)|$ of edges in tree T, we have $|V(G_{T,\mathcal{C}})| = |\mathcal{C}| = O(|E(T)|)$. The following lemma provides a bound for $|E(G_{T,\mathcal{C}})|$.

Lemma 10. *The number of edges in comigration graph $G_{T,\mathcal{C}}$ is at most the number of edges in T*, that is, $|E(G_{T,\mathcal{C}})| = O(|E(T)|)$.

Thus, by Lemma 10, TCC can be solved in $O(|V(G_{T,\mathcal{C}})| + |E(G_{T,\mathcal{C}})|) = O(|E(T)|)$ time if $G_{T,\mathcal{C}}$ is given. We still need to show how to construct the comigration graph $G_{T,\mathcal{C}}$ itself. One naive way to construct $G_{T,\mathcal{C}}$ is by checking each pair $(u, v), (u', v') \in M$ of migrations, and adding edge (C_s, C_t) to $G_{T,\mathcal{C}}$ if $(u, v) \in C_s$, $(u', v') \in C_t$, $v \preceq_T u'$, and there is no migration on the path from v to u'. But this approach requires quadratic time, so we propose a new linear-time algorithm. The recursive algorithm BuildComigrationGraph $(T, M, \mathcal{C}, v)$ takes as input a tree T, set M of migrations, set $\mathcal{C}$ of comigrations, and a vertex $v \in V(T)$. It returns two outputs: (i) a comigration graph denoted as $G_{T_v,\mathcal{C}}$ such that 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 such that $C \in X_s$ if C includes a migration (u', v') that is the first migration encountered on a directed path from v to any leaf. 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 in Appendix A.1.

Theorem 4. *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

In the previous section, we have shown that PCC is NP-hard. We solve the problem to optimality using an ILP. To solve the problem to optimality, we formulate an ILP, modeling comigrations $\mathcal{C}$ and timestamp labeling τ for the given set $M(T, \ell)$ of migrations induced by a given location labeling ℓ of a given tree T. The objective is to minimize the number $|\mathcal{C}|$ of comigrations while ensuring that $\mathcal{C}$ is spatiotemporally consistent.

4.2.1. Timestamp labeling. First, we begin by noting that the number of unique timestamps is at most the number $|M(T, \ell)|$ of migrations. Thus, we enumerate all possible timestamps as $\{1, \ldots, |M(T, \ell)|\}$. To model the assignment of timestamps $\tau((u, v))$ to migration edges $(u, v) \in M(T, \ell)$, we introduce binary variables $\mathbf{x} \in \{0, 1\}^{M(T, \ell) \times |M(T, \ell)|}$ such that $x_{(u,v),e}$ is 1 if $\tau((u, v)) = e$ and 0 otherwise. We have the following corresponding constraints, ensuring each migration edge is assigned one timestamp.

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

For any two migrations $(u, v), (u', v') \in M(T, \ell)$ where $v \preceq_T u'$, we require $\tau((u, v)) < \tau((u', v'))$ by the definition of temporal consistency (Definition 6). Now if $\tau((u, v)) < \tau((u', v'))$ then for any $\tau((u, v)) \leq E < \tau((u', v'))$ we have $\sum_{e=1}^{E} x_{(u,v),e} > \sum_{e=1}^{E} x_{(u',v'),e}$. Conversely, if $E < \tau((u, v))$ or $E \geq \tau((u', v'))$ then $\sum_{e=1}^{E} x_{(u,v),e} = \sum_{e=1}^{E} x_{(u',v'),e}$. We combine these two conditions to form the following constraints.

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

where $\pi(T, \ell)$ 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'. Note that the third condition is not necessary but results in fewer constraints, potentially speeding up the ILP.

4.2.2. 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 that have the same timestamp given by a timestamp labeling τ. In general, one may use the same timestamp for two comigrations occurring between distinct pairs of locations. However, in this formulation, we will require each comigration to have a unique timestamp, which we use to identify the comigration. This is without loss of generality because one can relabel any temporally consistent τ to use unique timestamps maintaining temporal consistency. Thus, to model comigrations, we introduce binary variables $\mathbf{x} \in \{0, 1\}^{|M(T,\ell)| \times \Sigma \times \Sigma}$, where $y_{e,s,t} = 1$ if there exists at least one migration (u, v) such that $\ell(u) = s$, $\ell(v) = t$, and $\tau((u, v)) = e$, and $y_{e,s,t} = 0$ otherwise. We have the following constraints ensuring that each timestamp corresponds to at most one comigration.

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

For each migration (u, v) with timestamp $\tau((u, v)) = e$, we force $y_{e,\ell(u),\ell(v)}$ to be 1 as follows.

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

4.2.3. Symmetry-breaking constraints. To increase performance, we use symmetry breaking constraints enforcing smaller timestamps to be used first.

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

4.2.4. Optimization function. Since we require each comigration to have a unique timestamp, the total number of comigrations equals the number of nonzero entries in y.

$$\min \sum_{e=1}^{|M(T,\ell)|} \sum_{s \in \Sigma} \sum_{t \in \Sigma} y_{e,s,t}.$$

Note that the objective function will ensure that $y_{e,s,s} = 0$ for all timestamps $e \in [|M(T, \ell)|]$ and $s \in \Sigma$.

4.2.5. Model size. PCC's ILP consists of $O(|M(T, \ell)|(|M(T, \ell)| + |\Sigma|^2)) = O(|E(T)|)^2$ variables and $O(|M(T, \ell)|^2) = O(|E(T)|^2)$ constraints.

4.3. ILP for the PCCH problem

In the previous section, we showed PCCH to be NP-hard. We solve the problem to optimality using an ILP. To do so, we must model (i) a location labeling ℓ, (ii) comigrations $\mathcal{C}$ identified by the labels of endpoints and timestamps of the member edges, (iii) an assignment of edges to parts, and (iv) symmetry-breaking constraints. The details of each step are discussed as follows.

4.3.1. Location labeling. To model location labeling ℓ, we introduce binary variables $\mathbf{z} \in \{0, 1\}^{V(T) \times \Sigma}$ such that $z_{v,s} = 1$ if $\ell(v) = s$, and $z_{v,s} = 0$ otherwise. As each vertex must be labeled by a location, we have

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

In addition, for the leaves of T, we force location labeling ℓ to match with input leaf labeling $\hat{\ell}$.

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

4.3.2. Timestamp labeling. For efficient ILP formulation, we assign timestamps on nonmigrations and include them in comigrations. This modification does not change the original PCCH algorithm, as the timestamps on nonmigrations can be ignored while still ensuring temporal consistency. Again the number of distinct comigrations and thus timestamps is at most the number $|E(T)|$ of edges, allowing us to enumerate our timestamps as $\{1, \ldots, |E(T)|\}$. Like our ILP for PCC, we introduce binary variables $\mathbf{x} \in \{0, 1\}^{E(T) \times \Sigma \times \Sigma \times |E(T)|}$ s.t. $x_{(u,v),s,t,e}$ is 1 if $\ell(u) = s$, $\ell(v) = t$, and $\tau((u, v)) = e$, and $x_{(u,v),s,t,e} = 0$ otherwise. These described conditions are enforced by the following three conditions.

$$\sum_{t\in\Sigma}\sum_{e=1}^{|E(T)|} x_{(u,v),s,t,e} \le z_{u,s}, \qquad \forall (u,v)\in E(T), \forall s\in\Sigma,$$

$$\sum_{s\in\Sigma}\sum_{e=1}^{|E(T)|} x_{(u,v),s,t,e} \le z_{v,t}, \qquad \forall (u,v)\in E(T), \forall t\in\Sigma,$$

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

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

$$\sum_{s\in\Sigma}\sum_{t\in\Sigma}\sum_{e=1}^{E} x_{(u,v),s,t,e} \ge \sum_{s\in\Sigma}\sum_{t\in\Sigma}\sum_{e=1}^{E} x_{(v,w),s,t,e}$$
$$\forall (u,v),(v,w)\in E(T), \forall E\in[|E(T)|]$$

4.3.3. Comigrations. Similar to our ILP for PCC, we again require each comigration to have a unique timestamp and use the timestamps to identify individual comigrations in this ILP. To that end, we introduce binary variables $y\in\{0,1\}^{|E(T)|\times\Sigma\times\Sigma}$ where $y_{e,s,t}=1$ if there exists a migration (u,v) such that $\ell(u)=s$, $\ell(v)=t$, and $\tau((u,v))=e$, and $y_{e,s,t}=0$ otherwise. The following constraint ensures spatial consistency by enforcing each comigration to be associated with a specific pair of locations.

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

For each edge (u,v) with $\ell(u)=s$, $\ell(v)=t$, and $\tau((u,v))=e$, we force $y_{e,s,t}$ to be 1.

$$x_{(u,v),s,t,e} \le y_{e,s,t}, \qquad \forall (u,v)\in E(T), \forall s,t\in\Sigma, \forall e\in[|E(T)|].$$

4.3.4. Symmetry-breaking constraints. Like the ILP model for PCC, we eliminate some symmetrical solutions by forcing smaller partition numbers to be used first.

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

4.3.5. Optimization function. We compute the number of migrations from variables x by counting the number of migrations. Since we ignore the comigrations with nonmigrations, we only count the number of comigrations that contain migrations from variables y. Thus, we define the objective function as

$$\min \sum_{(u,v)\in E(T)} \sum_{s,t\in\Sigma:s\ne t} \sum_{e=1}^{|E(T)|} x_{(u,v),s,t,e} + \frac{1}{|E(T)|}\sum_{e=1}^{|E(T)|}\sum_{s,t\in\Sigma:s\ne t} y_{e,s,t}.$$

In the optimization function, the factor $\frac{1}{|E(T)|}$ ensures that the ILP first minimizes the number of migrations and then the number of comigrations.

4.3.6. Model size. PCCH's ILP consists of $O(|E(T)|^2|\Sigma|^2)$ variables and $O(|E(T)|^2(|E(T)|^2|+|\Sigma|^2))=O(|E(T)|^4)$ constraints.

4.4. Workflow for inferring temporally consistent migration histories

MACHINA, like PCCH, employs an ILP for migration history inference. While both methods minimize migrations and comigrations lexicographically, MACHINA does not enforce temporal consistency like PCCH, resulting in a simpler ILP with $O(|E(T)|^2|\Sigma|)$ variables and $O(|E(T)|^2|\Sigma|)$ constraints, considerably fewer than PCCHs ILP with $O(E(T)^2|\Sigma|^2)$ variables and $O(|E(T)|^4)$ constraints. Due to the increased size of PCCHs ILP, we expect it to be slower compared to MACHINA. Furthermore, as per Proposition 1, MACHINA is guaranteed to infer optimal TCC when there is no reseeding. Therefore, we propose a workflow for migration history inference that leverages MACHINA's speed whenever feasible and ensures temporal consistency in the solutions by falling back to PCC and PCCH when necessary.

The workflow has five steps in total (Fig. 5). In step I, given an input tree T and leaf labeling $\hat{\ell}$, we run MACHINA to obtain a location labeling ℓ_{MACHINA} with the minimum number $\gamma(T, \ell_{\mathrm{MACHINA}})$ of compatible comigrations. For step II, we note that MACHINA does not explicitly output the comigrations. If one is not interested in this set of comigrations but only the number of comigrations, we can use Proposition 1 and check whether ℓ_{MACHINA} is reseeding-free, and if so, only report the number $\gamma(T, \ell_{\mathrm{MACHINA}})$ of comigrations. Therefore, we run Greedy-Comigrations to get the set of compatible comigrations $\mathcal{C}_{\mathrm{MACHINA}}$ such that $|\mathcal{C}_{\mathrm{MACHINA}}| = \gamma(T, \ell_{\mathrm{MACHINA}})$. In step III, we run the TCC algorithm to check whether $\mathcal{C}_{\mathrm{MACHINA}}$ is temporally consistent. If $\mathcal{C}_{\mathrm{MACHINA}}$ is temporally consistent then, by Corollary 1, $\mathcal{C}_{\mathrm{MACHINA}}$ is optimal and we terminate.

Otherwise if $\mathcal{C}_{\mathrm{MACHINA}}$ is temporally inconsistent, we proceed to step IV. In this step, we run the PCC ILP on input tree T and MACHINA location labeling ℓ_{MACHINA} to obtain the minimum set $\mathcal{C}_{\mathrm{PCC}}$ of TCC. Since Greedy-Comigrations does not guarantee $\mathcal{C}_{\mathrm{MACHINA}}$ to be temporally consistent, PCC helps checking whether there exists a temporally consistent set $\mathcal{C}_{\mathrm{PCC}}$ of comigrations such that $|\mathcal{C}_{\mathrm{PCC}}| = \gamma(T, \ell_{\mathrm{MACHINA}})$. If $|\mathcal{C}_{\mathrm{PCC}}| = \gamma(T, \ell_{\mathrm{MACHINA}})$ then the location labeling ℓ_{MACHINA} combined with the spatially consistent comigrations $\mathcal{C}_{\mathrm{PCC}}$ form an optimal solution to the PCCH (Corollary 1), thus allowing us to terminate the workflow. Otherwise, if $|\mathcal{C}_{\mathrm{PCC}}| > \gamma(T, \ell_{\mathrm{MACHINA}})$, we proceed with step V. In this final step, we run the PCCH ILP to compute the optimal location labeling ℓ_{PCCH} along with the minimum temporally consistent set $\mathcal{C}_{\mathrm{PCCH}}$ of comigrations.

5. RESULTS

In this section, we compare the performance of MACHINA with our methods on simulated (Section 5.1) and real 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, which uses Gurobi to solve the ILPs, as well as simulation and real data instances are available at https://github.com/elkebir-group/PCCH.

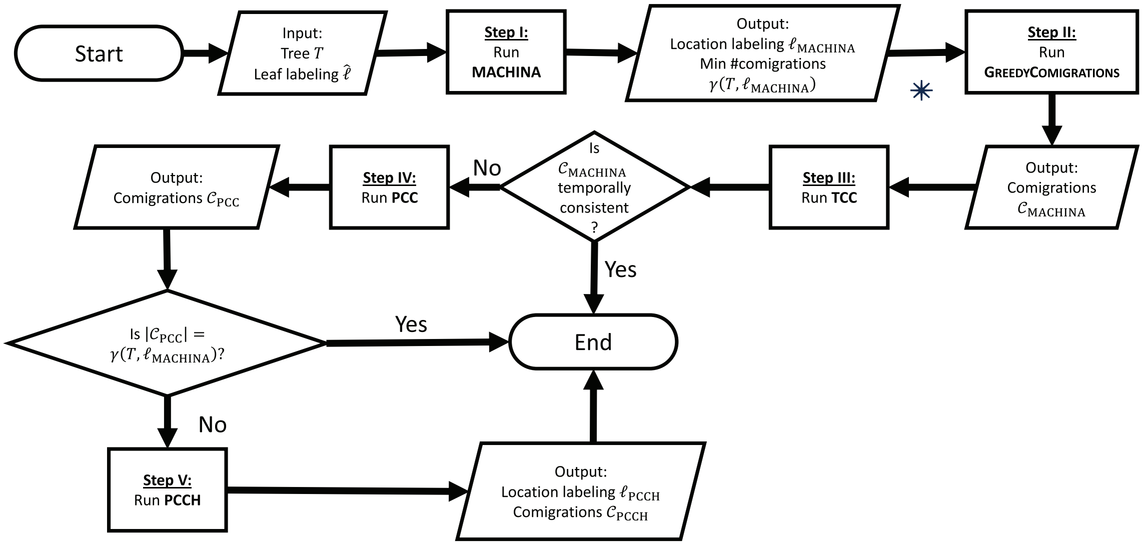
Start
Input: Tree T Leaf labeling $\hat{\ell}$
Step I: Run MACHINA
Output: Location labeling ℓ_{MACHINA} Min #comigrations $\gamma(T, \ell_{\mathrm{MACHINA}})$
*
Step II: Run GreedyComigrations
Output: Comigrations $\mathcal{C}_{\mathrm{MACHINA}}$
Step III: Run TCC
Is $\mathcal{C}_{\mathrm{MACHINA}}$ temporally consistent ?
No
Step IV: Run PCC
Output: Comigrations $\mathcal{C}_{\mathrm{PCC}}$
Yes
Is $|\mathcal{C}_{\mathrm{PCC}}| = \gamma(T, \ell_{\mathrm{MACHINA}})$?
Yes
End
No
Step V: Run PCCH
Output: Location labeling ℓ_{PCCH} Comigrations $\mathcal{C}_{\mathrm{PCCH}}$

FIG. 5. Workflow for inferring temporally consistent migration histories. The workflow consists of sequentially running MACHINA and the algorithms discussed in this article, falling back on more complex algorithms whenever necessary. *In case the user is not interested in the specific set $\mathcal{C}_{\mathrm{MACHINA}}$ of comigrations but only the number of comigrations, one can utilize Proposition 1 and check whether ℓ_{MACHINA} is reseeding-free, and if so, report the number $\gamma(T, \ell_{\mathrm{MACHINA}})$ of comigrations.

5.1. Simulated data

This section aims to evaluate the performance of our algorithms relative to MACHINA. To that end, we generated simulation instances following a three-step process. First, we sampled a comigration graph G resulting in a set $V(G) = \mathcal{C}$ of comigrations. Second, we sampled a tree T' with location labeling ℓ and assigned migrations $M(T', \ell)$ to the comigrations $\mathcal{C}$ such that T' and $\mathcal{C}$ induced the edges of the sampled comigration graph, that is, G is a subgraph of $G_{T,\mathcal{C}}$. We imposed an additional condition ensuring that each part of $\mathcal{C}$ consists of migrations that occur on distinct lineages. Third, we obtained the final tree T with leaf labeling $\hat{\ell}$ by adding edges to T' in a manner that minimizing the number of migrations and subsequently the number of compatible comigrations would yield the simulated comigrations $\mathcal{C}$.

We generated three classes of simulation instances, with increasing complexity in the initially sampled comigration graphs in the form of cycles. The details are provided in Supplementary Section D.1 and Supplementary Figure S3. We ran all five steps of the workflow for each instance without terminating prematurely. Thus, we ran MACHINA on each simulation instance $(T, \hat{\ell})$ resulting in a location labeling ℓ_{MACHINA}. We then used GREEDY-COMIGRATIONS to extract the set $\mathcal{C}_{\text{MACHINA}}$ of comigrations from $(T, \ell_{\text{MACHINA}})$. Next, we checked whether $\mathcal{C}_{\text{MACHINA}}$ was temporally consistent using the TCC algorithm. In addition, we ran the PCC algorithm on the output $(T, \ell_{\text{MACHINA}})$ produced by MACHINA, yielding a parsimonious set $\mathcal{C}_{\text{PCC}}$ of TCC. Finally, we ran the PCCH algorithm on the original simulation instance $(T, \hat{\ell})$ resulting in a location labeling ℓ_{PCCH} and set $\mathcal{C}_{\text{PCCH}}$ of comigrations. To assess the performance, we compared the outputs of each method, and also running times.

For our first set of simulations, we sampled five comigration graphs without any cycles, obtaining a total of five instances $(T, \hat{\ell})$, one for each sampled comigration graph. These instances had 26 to 74 vertices and included 3 to 7 locations. We expect all methods to yield temporally consistent solutions with identical numbers of migration and comigrations for these instances. Indeed, for each instance, we observed that $|M(T, \ell_{\text{MACHINA}})| = |M(T, \ell_{\text{PCCH}})|$, $|\mathcal{C}_{\text{MACHINA}}| = |\mathcal{C}_{\text{PCC}}| = |\mathcal{C}_{\text{PCCH}}|$, and that MACHINA's solution was temporally consistent (Fig. 6a). Note that the $\mathcal{C}_{\text{PCC}}$ and $\mathcal{C}_{\text{PCCH}}$ are by definition temporally consistent. In terms of running time, MACHINA outperformed PCCH slightly, with median running times of 12.029 seconds for MACHINA and 18.806 seconds for PCCH (Fig. 6b). Despite PCCs NP-hardness, the corresponding ILP executed much faster than MACHINA and PCCH due to fewer constraints and variables in the ILP model, with a median running time of 0.043 seconds (Fig. 6b and Supplementary Table S4).

We also executed our workflow on all five instances, which terminated at step III because of $\mathcal{C}_{\text{MACHINA}}$ being temporally consistent (Supplementary Table S4). As the workflow skipped steps IV and V (PCC and PCCH), and the combined running time for steps II and III (GREEDY-COMIGRATIONS and TCC) was significantly shorter, with a median of 0.002 seconds, the workflow's running time closely matched that of MACHINA (Fig. 6b and Supplementary Table S4).

To generate the second set of simulation instances, we picked comigration graphs with $k \in \{1, 2, 3, 4\}$ disjoint cycles. For each value of k, we generated five comigration graphs with k cycles and simulated five instances $(T, \hat{\ell})$, totaling 20 instances. The simulated trees had 26 to 88 vertices and 3 to 11 locations. Because of the presence of cycles in the initially sampled comigration graphs, MACHINA failed to return a temporally consistent set $\mathcal{C}_{\text{MACHINA}}$ of comigrations for all the instances (Fig. 6a and Supplementary Table S4). As such, the number of comigrations inferred by MACHINA, PCC, and PCCH differed, although the number of migrations inferred by MACHINA matched that of PCCH. To be more specific, for the instances generated from initially sampled comigration graphs with $k \in \{1, 2, 3, 4\}$ cycles, MACHINA underestimated the minimum number of comigrations by k, that is, $|\mathcal{C}_{\text{MACHINA}}| = |\mathcal{C}_{\text{PCC}}| - k$ (Supplementary Table S4).

Note that MACHINA's inability to accurately determine the number of comigrations for a specific instance does not necessarily imply that the associated location labeling is incorrect. For example, in 9 out of 20 cases, ℓ_{MACHINA} matched ℓ_{PCCH}, and $\mathcal{C}_{\text{PCC}}$ computed from ℓ_{MACHINA} inferred by PCC matched $\mathcal{C}_{\text{PCCH}}$ computed by PCCH. But in the other 11 cases, $|\mathcal{C}_{\text{PCC}}|$ was greater than $|\mathcal{C}_{\text{PCCH}}|$, indicating that achieving the minimum comigration count with ℓ_{MACHINA} was not possible, rendering ℓ_{MACHINA} suboptimal (Fig. 6a and Supplementary Table S4). In these cases, we observed 1 to 3 vertices to be labeled differently between ℓ_{MACHINA} and ℓ_{PCCH}.

In Figure 6, we present a simulation instance with $k = 2$ cycles where MACHINA and PCCH produced different results. This instance corresponds to a tree T with 36 vertices and 5 locations. MACHINA provided the location labeling ℓ_{MACHINA} shown in Figure 6c, reporting 15 migrations and 9 comigrations.

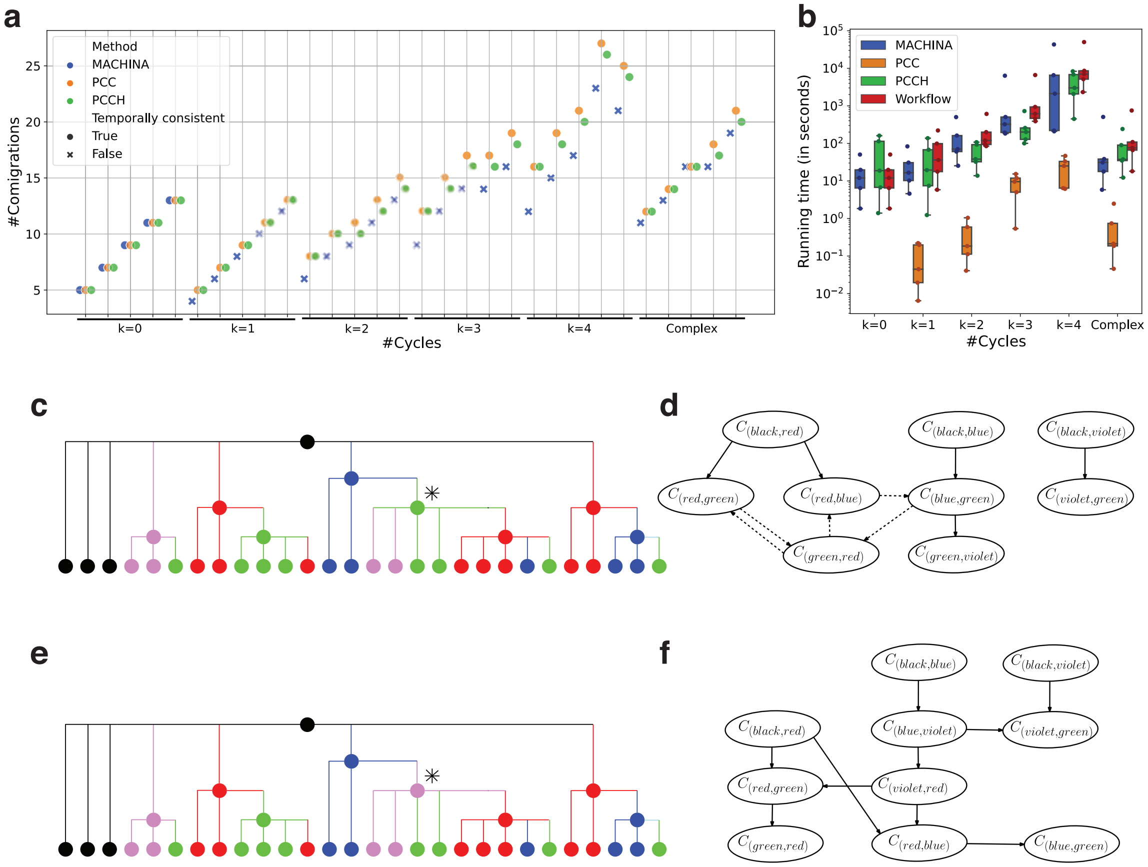

FIG. 6. Simulation results. **(a)** The inferred numbers of comigrations (y-axis) for each method (color) across simulation instances (x-axis), additionally indicating temporal consistency (shape). **(b)** The running time (y-axis) for each method (color) across simulation instances (x-axis). **(c–f)** One simulation instance $(T, \hat{\ell})$ where MACHINA fails to return a temporally consistent solution is included here, with **(c)** showing the MACHINA location labeling ℓ_{MACHINA} and **(d)** the corresponding comigration graph $G_{T,\mathcal{C}_{\text{MACHINA}}}$ containing several cycles (dashed). **(e)** By contrast, PCCH infers a location labeling ℓ_{PCCH} that differs at the indicated vertex ("*") and TCC $\mathcal{C}_{\text{PCCH}}$, **(f)** not containing any cycles in the induced comigration graph $G_{T,\mathcal{C}_{\text{PCCH}}}$. Note that while $|M(T, \ell_{\text{MACHINA}}| = |M(T, \ell_{\text{PCCH}}| = 15$ for both solutions, we have $|\mathcal{C}_{\text{MACHINA}}| = 9 < 10 = |\mathcal{C}_{\text{PCCH}}|$.

However, the corresponding comigration graph $G_{T,\mathcal{C}_{\text{MACHINA}}}$ in Figure 6d revealed two disjoint cycles, indicating temporal inconsistency in MACHINA's comigrations (Theorem 1). Running PCC on the location labeling ℓ_{MACHINA} inferred by MACHINA deduced the minimum set $\mathcal{C}_{\text{PCC}}$ of TCC to be of size 11. Conversely, PCCHs location labeling ℓ_{PCCH}, depicted in Figure 6e, accounted for 15 migrations and 10 comigrations, with the corresponding comigration graph $G_{T,\mathcal{C}_{\text{PCCH}}}$ in Figure 6f being a DAG. So the location labeling ℓ_{PCCH} minimizes the number of TCC, and the solution returned by MACHINA is temporally inconsistent and suboptimal.

Although MACHINA was faster in $k = 1$ cases (median: 16.56 seconds for MACHINA, 19.48 seconds for PCCH), a clear pattern does not emerge for the instances where $k > 1$ (Fig. 6b and Supplementary Table S4). For instance, MACHINA was slower for $k = 3$ instances (median: 323.448 seconds for MACHINA, 201.011 seconds for PCCH), but faster for $k = 4$ instances (median: 2109.8 seconds for MACHINA, 3001.178 seconds for PCCH). Since MACHINA returned temporally inconsistent comigrations $\mathcal{C}_{\text{MACHINA}}$ this time, the workflow ran both PCC and PCCH and terminated at step V. The workflow's running time was primarily influenced by MACHINA and PCCH, as the running times of GREEDY-COMIGRATIONS, TCC, and PCC were negligible in comparison.

Finally, we constructed our third set of simulations by sampling comigration graphs with complex, nested cycles. Specifically, we began by sampling a comigration graph with one cycle. Then, we randomly selected pairs of vertices from the comigration graph, ensuring that they do not share an edge with the cycle, and connected them. We generated five such comigration graphs, and for each of these comigration graph, we simulated one tree T with leaf labeling $\hat{\ell}$ following the aforementioned simulation procedure. The simulation instances had 37 to 61 vertices and 7 to 10 locations. Like the previous case, MACHINA returned a temporally inconsistent set $\mathcal{C}_{\text{MACHINA}}$ of comigrations for all the simulation instances (Fig. 6a and Supplementary Table S4). The differences between the number of comigrations reported by MACHINA and PCC were between 1 and 2, and for two instances, MACHINA failed to return the optimal location labeling, that is, $|\mathcal{C}_{\text{PCC}}| > |\mathcal{C}_{\text{PCCH}}|$ (Fig. 6a and Supplementary Table S4).

In terms of running time, we observed MACHINA outpacing PCCH, with a median running time of 31.176 seconds for MACHINA and 37.542 seconds for PCCH (Fig. 6b and Supplementary Table S4). Like the second class of simulations, the workflow terminated at step V, and the running time was dominated by MACHINA and PCCH.

5.2. *Real data*

5.2.1. Ovarian cancer. We applied PCCH to infer the migration history of seven patients diagnosed with high-grade serous ovarian cancer from McPherson et al. (2016). McPherson et al. (2016) sequenced 68 tumor samples across seven patients, encompassing samples from various sites such as the ovary, omentum, fallopian tube, peritoneal locations, and distant metastatic sites, using whole genome and targeted sequencing. After identifying the dominant clones from detected SNVs and rearrangement breakpoints, they constructed clone trees T using a probabilistic phylogenetic model based on the stochastic Dollo process. Finally, for each patient, they inferred the migration history by finding the location labeling ℓ minimizing only the number $|M(T, \ell)|$ of migrations. El-Kebir et al. (2018) reanalyzed the same dataset using MACHINA and identified simpler migration patterns for patients 1, 3, and 9 based on the comigration criterion.

For instance, for patient 1, McPherson et al. (2016) originally identified the right ovary (ROv) as the primary tumor location, as their reported optimal location labeling had 13 migrations and 10 comigrations with ROv as the primary site. Also, they reported the occurrence of metastasis-to-metastasis migration for patient 1. In contrast, MACHINA found a more optimal solution with the same number of migrations but only seven comigrations, designating the left ovary as the primary tumor location. Furthermore, MACHINA inferred a simpler migration pattern for patient 1 without reporting any metastasis-to-metastasis migration.

For each of the seven patients, we generated the location labeling with timestamps by solving PCCH. We found that PCCHs location labelings perfectly matched those of MACHINA. Moreover, we found both methods returned the same number of comigrations. As both the location labelings and the number of comigrations matched, MACHINA's solutions are temporally consistent. As an example, we show the PCCH output for patient 1 in Figure 7a with location and timestamp labels. Both MACHINA and PCCH

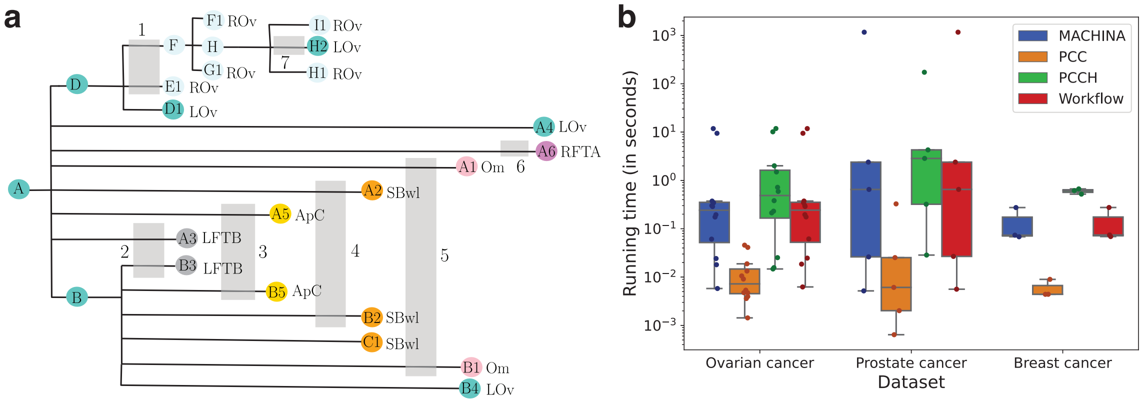

FIG. 7. MACHINA and PCCH results for ovarian (McPherson et al., 2016), prostate (Gundem et al., 2015), and breast cancer (Hoadley et al., 2016) datasets. **(a)** PCCH results for ovarian cancer patient 1. Migrations enclosed within the same gray box represent a comigration (additionally labeled by timestamp) and vertex colors specify location labeling. **(b)** The running time (y-axis) for each method (color) across real datasets (x-axis).

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.

We show the running time analysis for PCC, PCCH, MACHINA, and the workflow in Figure 7b and Supplementary Table S1. We found that PCCH generally takes slightly longer to finish (median of 0.474 seconds vs. 0.244 seconds for MACHINA). This is expected, as unlike MACHINA, PCCH includes checks for temporal consistency and returns timestamps along with a location labeling. Similar to the findings on simulated data, we found PCC to be significantly faster than PCCH or MACHINA. Since the MACHINA comigrations are temporally consistent, the workflow stops at step III, resulting in the running time of the workflow matching closely with that of MACHINA.

5.2.2. Prostate cancer. We ran PCCH and inferred the migration history of five androgen-deprived metastaic prostate cancer patients from Gundem et al. (2015). For the five selected patients, Gundem et al. (2015) sequenced both primary (prostate) and metastatic samples using whole-genome sequencing (WGS) technology. For each patient, they constructed a clone tree T by first identifying mutation clusters and calculating cancer cell fractions of each cluster in each sample by using an n-dimensional Bayesian Dirichlet process, and then inferring evolutionary relationships between pairs of mutation clusters by applying the ''pigeon-hole'' principle to mutation clusters within individual samples. To infer the migration histories, they deduced the location of origin of each mutation cluster by examining cancer cell fractions in each sample and using the ''pigeon-hole'' principle, and reported metastasis-to-metastasis migration in four (A10, A22, A31, and A32) out of five patients in consideration.

The samples from the same five patients were reanalyzed by MACHINA in El-Kebir et al. (2018), where it found simpler solutions with metastasis-to-metastasis spread only in two patients (A22, A32). MACHINA also did not report reseeding for any of the patients, which implies that the migration histories inferred by MACHINA are temporally consistent by Proposition 1. Indeed, we found that the inferred location labeling and the number of comigrations were identical for both PCCH and MACHINA. In terms of running times, we observed similar trends (Fig. 7b and Supplementary Table S2)—MACHINA was slightly faster than PCCH (median of 27.795 seconds vs. 0.67 seconds for MACHINA), although for patient A22, MACHINA (1702.24 seconds) needed more time than PCCH (185.18 seconds). For PCC, the running time was significantly shorter (median: 0.025 seconds). The workflow stops at step III because of $\mathcal{C}_{\text{MACHINA}}$ being temporally consistent,

5.2.3. Breast cancer. We applied our methods to examine the migration history of two triple-negative breast cancer patients from Hoadley et al. (2016). DNA whole-genome sequencing was conducted on matched primary and multiple distant metastasis samples for both patients. The clonal structure was inferred using SciClone (Miller et al., 2014), and the phylogeny was determined using the ClonEvol R package (Dang et al., 2017). For patient A1, ClonEvol reported two potential clone trees due to its inability to accurately determine the evolutionary origin of clone 7. For patient A1, MACHINA recapitulated the findings reported in Hoadley et al. (2016) that all the clones except clones 6 and 9 originated in the primary location for both trees. For patient A7, MACHINA reported a parsimonious solution with eight migrations and six comigrations, and a comigration from primary location to lung for clones 2 and 4, which agreed with Hoadley et al. (2016).

All the results returned by MACHINA were temporally consistent, and so the workflow stopped at step III. Consequently, the migration histories inferred by MACHINA and PCCH were identical. Running times followed the same trend (Fig. 7b and Supplementary Table S3), with MACHINA being slightly faster than PCCH (median of 0.613 seconds vs. 0.074 seconds for MACHINA), and PCC being the fastest (median: 0.004 seconds).

6. CONCLUSION

In this article, we addressed a flaw in the definition of comigration adopted by MACHINA (El-Kebir et al., 2018). Specifically, we precisely defined spatial and temporal consistency for comigrations, leading to the formulation of three successive problems. The first problem, TCC, determines temporal consistency given a set of comigrations and derives a timestamp labeling for migrations in case the comigrations are temporally consistent. We showed that TCC can be solved in linear time. The second problem, PCC infers the smallest set of TCC given the locations of both leaf and internal vertices. We proved the problem to be NP-hard, indicating that even if the location of origin of every vertex and thus every migration is given as input, it is still computationally hard to deduce which migrations occurred simultaneously under a parsimony criterion.

Our third problem, PCCH, takes as input a leaf labeling, and infers the location labeling that minimizes the number of migrations, and subsequently the number of spatiotemporally consistent comigrations. We proved that PCCH is also NP-hard. In addition, we discussed MACHINA's views on comigrations and its limitations concerning temporal consistency and reported a sufficient condition under which MACHINA accurately computes comigrations. We presented ILP models for PCC and PCCH and proposed a workflow that combines the strengths of MACHINA, PCC, and PCCH—by using TCC to verify MACHINA's results and resorting to PCC and PCCH when needed. Finally, we conducted a comparative analysis of PCCH and MACHINA's performance on simulated and real data.

We generated simulation instances to investigate when MACHINA fails to determine temporally consistent comigrations and showed that MACHINA underestimates comigrations and may yield suboptimal location labeling in the presence of comigration graph cycles. For real data, PCCH returned the same location labeling as MACHINA for all instances.

PCCH offers several promising avenues for future research. While our current study focused on applying PCCH exclusively to cancer data, its versatility extends to inferring migration history in various organisms, including disease pathogens, as discussed earlier. Broadening the application of PCCH to diverse real datasets is crucial for gaining a comprehensive understanding of temporal inconsistency in practical scenarios. Drawing inspiration from MACHINA, which introduced parsimonious migration history with tree refinement, we plan to expand PCCH to incorporate tree refinement, aiming to minimize the number of migrations and comigrations lexicographically across all location labelings for possible tree refinements of the input tree. Furthermore, a captivating challenge lies in exploring the existence of multiple optimal solutions within the PCCH framework. Currently, PCCH provides a single optimal solution, yet, instances may arise where distinct location labelings yield the same number of migrations and TCC. Investigating the solution space within PCCH to detect and characterizing these alternatives represents a promising avenue for future research in this field.

ACKNOWLEDGMENTS

An earlier version of this article was published in WABI 2023 (doi: 10.4230/LIPIcs.WABI.2023.9). This project started as a collaboration at the Computational Genomics Summer Institute 2022.

AUTHORS' CONTRIBUTIONS

M.S.R.: Conceptualization, Implementation, Formal analysis, and Writing—Review. S.S.: Conceptualization and Writing—review. M.E.-K.: Conceptualization, Validation, and Writing—review and editing.

AUTHOR DISCLOSURE STATEMENT

The authors declare they have no conflicting financial interests.

FUNDING INFORMATION

Mohammed El-Kebir was supported by the National Science Foundation (CCF-2046488) as well as funding from the Cancer Center at Illinois. Sagi Snir was supported by the Israel Science Foundation (Grant No. ISF 1927/21) and the American/Israeli Binational Science Foundation (Grant No. BSF 2021139).

SUPPLEMENTARY MATERIAL

Supplementary Data
Supplementary Table S1
Supplementary Table S2
Supplementary Table S3
Supplementary Table S4
Supplementary Figure S1
Supplementary Figure S2
Supplementary Figure S3

REFERENCES

Aceto N, Bardia A, Miyamoto DT, et al. Circulating tumor cell clusters are oligoclonal precursors of breast cancer metastasis. Cell 2014;158(5):1110–1122.

Birkbak NJ, McGranahan N. Cancer genome evolutionary trajectories in metastasis. Cancer Cell 2020;37(1):8–19.

Campbell F, Cori A, Ferguson N, Jombart T. Bayesian inference of transmission chains using timing of symptoms, pathogen genomes and contact data. PLoS Comput Biol 2019;15(3):e1006930.

Chauve C, Rafiey A, Davin AA, et al. MaxTiC: Fast ranking of a phylogenetic tree by maximum time consistency with lateral gene transfers. bioRxiv 2017;2017:127548.

Cheung KJ, Ewald AJ. A collective route to metastasis: Seeding by tumor cell clusters. Science 2016;352(6282):167–169.

Cheung KJ, Padmanaban V, Silvestri V, et al. Polyclonal breast cancer metastases arise from collective dissemination of keratin 14-expressing tumor cell clusters. Proc Natl Acad Sci U S A 2016;113(7):E854–E863.

Comen E, Norton L, Massague J. Clinical implications of cancer self-seeding. Nat Rev Clin Oncol 2011;8(6):369–377.

Dadiani M, Kalchenko V, Yosepovich A, et al. Real-time imaging of lymphogenic metastasis in orthotopic human breast cancer. Cancer Res 2006;66(16):8037–8041.

Dang H, White B, Foltz S, et al. Clonevol: clonal ordering and visualization in cancer sequencing. Ann Oncol 2017;28(12):3076–3082.

David LA, Alm EJ. Rapid evolutionary innovation during an Archaean genetic expansion. Nature 2011;469(7328): 93–96.

Dellicour S, Baele G, Dudas G, et al. Phylodynamic assessment of intervention strategies for the West African Ebola virus outbreak. Nat Commun 2018;9(1):2222.

El-Kebir M. Parsimonious Migration History Problem: Complexity and Algorithms. In: 18th International Workshop on Algorithms in Bioinformatics (WABI 2018). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik: Germany; 2018.

El-Kebir M, Satas G, Raphael BJ. Inferring parsimonious migration histories for metastatic cancers. Nat Genet 2018;50(5):718–726.

Faries MB, Steen S, Ye X, et al. Late recurrence in melanoma: Clinical implications of lost dormancy. J Am Coll Surg 2013;217(1):27–34.

Faye O, Boëlle P-Y, Heleze E, et al. Chains of transmission and control of Ebola virus disease in Conakry, Guinea, in 2014: An observational study. Lancet Infect Dis 2015;15(3):320–326.

Ferguson NM, Donnelly CA, Anderson RM. Transmission intensity and impact of control policies on the foot and mouth epidemic in Great Britain. Nature 2001;413(6855):542–548.

Gundem G, Van Loo P, Kremeyer B, et al. The evolutionary history of lethal metastatic prostate cancer. Nature 2015;520(7547):353–357.

Hoadley KA, Siegel MB, Kanchi KL, et al. Tumor evolution in two patients with basal-like breast cancer: A retrospective genomics study of multiple metastases. PLoS Med 2016;13(12):e1002174.

Kahn AB. Topological sorting of large networks. Commun ACM 1962;5(11):558–562.

Kok SY, Oshima H, Takahashi K, et al. Malignant subclone drives metastasis of genetically and phenotypically heterogenous cell clusters through fibrotic niche generation. Nat Commun 2021;12(1):863.

Lafond M, Hellmuth M. Reconstruction of time-consistent species trees. Algorithms Mol Biol 2020;15(1):1–27.

Libeskind-Hadas R, Charleston MA. On the computational complexity of the reticulate cophylogeny reconstruction problem. J Comput Biol 2009;16(1):105–117.

Maddipati R, Stanger BZ. Pancreatic cancer metastases harbor evidence of polyclonality. Cancer Discov 2015; 5(10):1086–1097.

Margeridon-Thermet S, Shulman N, Ahmed A, et al. Ultra-deep pyrosequencing of hepatitis B virus quasispecies from nucleoside and nucleotide reverse-transcriptase inhibitor (NRTI)–treated patients and nrti-naive patients. J Infect Dis 2009;199(9):1275–1285.

Marrinucci D, Bethel K, Kolatkar A, et al. Fluid biopsy in patients with metastatic prostate, pancreatic and breast cancers. Phys Biol 2012;9(1):016003.

McPherson A, Roth A, Laks E, et al. Divergent modes of clonal spread and intraperitoneal mixing in high-grade serous ovarian cancer. Nat Genet 2016;48(7):758–767.

Merkle D, Middendorf M. Reconstruction of the cophylogenetic history of related phylogenetic trees with divergence timing information. Theory Biosci 2005;123:277–299.

Miller CA, White BS, Dees ND, et al. Sciclone: Inferring clonal architecture and tracking the spatial and temporal patterns of tumor evolution. PLoS Comput Biol 2014;10(8):e1003665.

Nøjgaard N, Geiß M, Merkle D, et al. Time-consistent reconciliation maps and forbidden time travel. Algorithms Mol Biol 2018;13(1):1–17.

Rambaut A, Posada D, Crandall K, et al. The causes and consequences of HIV evolution. Nat Rev Genet 2004;5(1): 52–61.

Räihä K-J, Ukkonen E. The shortest common supersequence problem over binary alphabet is NP-complete. Theor Comput Sci 1981;16(2):187–198; doi: 10.1016/0304-3975(81)90075-X

Sanborn JZ, Chung J, Purdom E, et al. Phylogenetic analyses of melanoma reveal complex patterns of metastatic dissemination. Proc Natl Acad Sci U S A 2015;112(35):10995–11000.

Sashittal P, El-Kebir M. SharpTNI: Counting and sampling parsimonious transmission networks under a weak bottleneck. bioRxiv 2019;2019:842237.

Sashittal P, El-Kebir M. Sampling and summarizing transmission trees with multi-strain infections. Bioinformatics 2020;36(Suppl 1):i362–i370.

Slatkin M, Maddison WP. A cladistic measure of gene flow inferred from the phylogenies of alleles. Genetics 1989;123(3):603–613.

Sobel Leonard A, Weissman DB, Greenbaum B, et al. Transmission bottleneck size estimation from pathogen deep-sequencing data, with an application to human influenza A virus. J Virol 2017;91(14):e00171–17.

Somarelli JA, Ware KE, Kostadinov R, et al. Phylooncology: Understanding cancer through phylogenetic analysis. Biochim Biophys Acta 2017;1867(2):101–108.

Spada E, Sagliocca L, Sourdis J, et al. Use of the minimum spanning tree model for molecular epidemiological investigation of a nosocomial outbreak of hepatitis C virus infection. J Clin Microbiol 2004;42(9):4230–4236.

Tabassum DP, Polyak K. Tumorigenesis: It takes a village. Nat Rev Cancer 2015;15(8):473–483.

Tofigh A, Hallett M, Lagergren J. Simultaneous identification of duplications and lateral gene transfers. IEEE/ACM Trans Comput Biol Bioinformatics 2010;8(2):517–535.

Tonkin-Hill G, Martincorena I, Amato R, et al. Patterns of within-host genetic diversity in SARs-CoV-2. Elife 2021;10:e66857.

Wang G, Sherrill-Mix S, Chang K, et al. Hepatitis C virus transmission bottlenecks analyzed by deep sequencing. J Virol 2010;840(12):6218–6228.

Yamamoto A, Doak AE, Cheung KJ. Orchestration of collective migration and metastasis by tumor cell clusters. Annu Rev Pathol 2023;18:231–256.

Yu M, Bardia A, Wittner BS, et al. Circulating breast tumor cells exhibit dynamic changes in epithelial and mesenchymal composition. Science 2013;339(6119):580–584.

Address correspondence to:
Dr. Mohammed El-Kebir
Department of Computer Science
University of Illinois Urbana-Champaign
201 North Goodwin Avenue
Urbana, IL 61801-3028
USA

E-mail: melkebir@illinois.edu